GA approach for finding Rough Set decision rules based on bireducts

Oleg Rybkin

Department of Computer Science

Supervisors:
Prof. Beatrice Ombuki-Berman
Prof. Ivo Düntsch

Submitted in partial fulfillment
of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

Abstract

Feature selection plays an important role in knowledge discovery and data mining nowadays. In traditional rough set theory, feature selection using reduct - the minimal discerning set of attributes - is an important area. Nevertheless, the original definition of a reduct is restrictive, so in one of the previous research it was proposed to take into account not only the horizontal reduction of information by feature selection, but also a vertical reduction considering suitable subsets of the original set of objects. Following the work mentioned above, a new approach to generate bireducts using a multi–objective genetic algorithm was proposed. Although the genetic algorithms were used to calculate reduct in some previous works, we did not find any work where genetic algorithms were adopted to calculate bireducts. Compared to the works done before in this area, the proposed method has less randomness in generating bireducts. The genetic algorithm system estimated a quality of each bireduct by values of two objective functions as evolution progresses, so consequently a set of bireducts with optimized values of these objectives was obtained. Different fitness evaluation methods and genetic operators, such as crossover and mutation, were applied and the prediction accuracies were compared. Five datasets were used to test the proposed method and two datasets were used to perform a comparison study. Statistical analysis using the one-way ANOVA test was performed to determine the significant difference between the results. The experiment showed that the proposed method was able to reduce the number of bireducts necessary in order to receive a good prediction accuracy. Also, the influence of different genetic operators and fitness evaluation strategies on the prediction accuracy was analyzed. It was shown that the prediction accuracies of the proposed method are comparable with the best results in machine learning literature, and some of them outperformed it.

Acknowledgements

I would like to thank the following people and organizations for their support during my research. This thesis would have been impossible without them:

- Beatrice Ombuki-Berman and Ivo Düntsch for all their support and inspiration. Thank you for being amazing supervisors.
- My parents, Sergey and Natalya, my brother, Denis and my grandmother Maria for being there for me at all times, supporting and providing me with the necessary resources to accomplish my studies.
- Brian Ross and Michael Winter for support and guidance.
- The Department of Computer Science for their help and support.

O.R.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Feature selection or attribute selection is a term commonly used in data mining to describe the methods and techniques used for reducing the number of inputs to a manageable size in order to save computational time and not analyze irrelevant or less relevant variables (features, attributes). Nowadays, attribute subset selection plays an important role in knowledge discovery. In the modern world, where much of different information is stored in datasets, efficient algorithms for attribute selection and data classification are widely used.

In the traditional rough set theory, feature selection through the use of reducts - the minimal discerning sets of attributes - has been an important area. As finding reducts is computationally unfeasible [27], many methods of finding feature sets with an acceptable quality of classification have been proposed within the classical rough set theory, see for example, [2, 20, 28].

As mentioned above, the problem of finding minimal subset of discerning attributes is not trivial and NP-complete [27]. Many feature selection algorithms are designed to deal with datasets that have a small number of attributes and/or objects. Therefore, they cannot provide a result for a huge number of attributes and/or objects in feasible time.

Recently, Ślęzak and Janusz [30] proposed to take into account not only the horizontal reduction of information by feature selection, but also a vertical reduction by considering suitable subsets of the original set of objects. Thus, the aim is to find areas in the two-dimensional object $\times$ attribute plane which are best suited for classification in some sense. This leads to the following definition: A *(decision) bireduct* is a pair $\langle B, X\rangle$ such that $B \subseteq A, X \subseteq U$ and

R1. For all $b \in B, x, y \in X$, $b(x) = b(y)$ implies $d(x) = d(y)$; in this case, we write $B \Rrightarrow_X d$. (B is discerning all elements of X)

R2. If $C \subsetneq B$, there are $x, y \in X$ such that $c(x) = c(y)$ for all $c \in C$ and $d(x) \neq d(y)$. (Minimality of B with respect to X)

R3. If $X \subsetneq Y$, there are $x, y \in Y$ such that $b(x) = b(y)$ for all $b \in B$ and $d(x) \neq d(y)$. (Maximality of X with respect to B)

Decision rules now can be obtained as in the classical rough set theory by restricting the scope of quantifiers to parts of a bireduct $\langle B, X \rangle$. Bireducts – which are consistent on their object set – may be viewed as inducing approximate rules on the whole object set U.

As finding (optimal) bireducts is NP – hard [33], obtaining optimal solutions for this kind of problem is computationally intractable. Thus heuristic methods are required for bireduct discovery in a realistic amount of time. In our research genetic algorithms are our method of choice for finding bireducts.

Even though the genetic algorithms were mentioned as a possible way of calculating reducts [25] and used to calculate reduct in a few previous works [4, 35], we did not find any work where genetic algorithms were adopted to calculate bireducts.

Compared to the work by Ślęzak and Janusz [30], our method has less randomness in generating bireducts. In our case we estimate a quality of each bireduct by values of two objective functions as evolution progresses, so, consequently we obtain a set of bireducts with optimized values of these objectives.

1.1 Goal

The goal of this thesis is to implement a multi-objective genetic algorithm (GA) approach and to make it suitable for finding bireducts. Different crossover and mutation operators were incorporated in this study and an empirical study carried out. Another aim is to try different fitness evaluation methods such as weighted-sum, normalized sum of ranks and Pareto ranking and compare their outcome. The final goal is based on all the experiments described above, namely to answer to the question, “can we use rules generated based on the bireducts for prediction?”.

1.2 Thesis Structure

The thesis is organized as follows: Chapter 2 provides background information about machine learning, genetic algorithms, multi-objective optimization, rough sets, reducts

and bireducts. Chapter 3 describes the implemented systems and algorithms in details. Chapter 4 provides experimental details and presents experimental discussions for each experiment. Lastly, the concluding remarks and future work is provided in Chapter 5.

Chapter 2

Background

This chapter provides the background information about the problem of finding bireducts (horizontal and vertical reduction) and rough sets decision rules generation. Also this chapter presents the background information about rough sets, reducts and bireducts, genetic algorithms, fitness evaluation methods and genetic operators used in our research.

2.1 Horizontal and vertical reduction and rough set decision rules generation

In many instances machine learning involves processing information characterized by a massive amount of features. However, it is possible to narrow down the quantity of features to a more manageable set without compromising the validity of label prediction. This technique is called feature selection (also - subset selection or attribute selection). The principal advantages of using feature selection prior to executing a learning algorithm are as follows:

1. Economy, lean data collection gathering fewer types of data typically means less time and money spent in the physical world. Feature selection can save time on measuring non-selected attributes. For instance, determining a group of crucial factors for biochip analysis and excluding a multitude of less relevant ones should be translate into savings in laboratory equipment, supplies and the quantity of tests.

2. Reduced computational complexity, streamlined processing choosing a relatively small subset of data features greatly reduces the volume of necessary com-

putations, which otherwise may become challenging. Many algorithms become computationally intractable in the presence of a large number of attributes, so feature selection can reduce the computational time in both training and prediction steps.

3. Enhanced generalization by reducing overfitting leaving out irrelevant and less relevant variables dramatically decreases the amount of information noise, enhancing the performance of learning algorithms.

4. Big picture insight, problem understanding narrowing down a set of features can lead to more efficient understanding of the problem.

While there are other ways to reduce the quantity of information features in massive data sets, only feature selection allows avoiding intensive data processing before the actual reduction takes place. This is especially important at the data collection stage. The added value of gaining the big picture knowledge in the process of feature selection is another unique benefit of this method.

The concept at the core of feature selection is the relevance of a given feature. Feature selection or filtering-out are based on the relevance of the features. Despite the apparent simplicity of this, there are a number of approaches to feature relevance. According to Gennari *et al.* [11], a feature is relevant if the conditional distribution of the labels is affected by it. It is also noteworthy that any useful and relevant feature may lose in significance depending on the nature of other features involved.

Since the accuracy of prediction is the main objective, feature selection can be defined as the process of choosing a much smaller quantity of data features out of massive quantity, that would allow acceptance accuracy of label prediction.

There are many different methods and algorithms that exist to perform feature selection. The one we use in our research is based on rough sets (RS) theory. One of the main advantages of RS is that it does not need any preliminary and/or additional knowledge about data compared to grade of membership in the fuzzy set theory or probability in statistics [24].

In rough sets, the minimal subset of attributes that preserves the partitioning of the universe and, hence, the ability to perform classification as the whole original attribute set is called a reduct. In some huge datasets, finding reducts is complicated because it might be impossible to reduce the number of attributes based on provided data. Using an approximate reduct, which ignores the attribute(s) if it discerns just a small number of objects has some issues [29, 31]. One of the ideas of how to deal with it was proposed in [30]. The idea does not just concentrate on a subset of attributes,

but rather concentrates on a subset of attributes that describe decisions, as well as on a subset of objects, for which this subset of attributes is valid.

2.2 Genetic Algorithms

A genetic algorithm (GA) is a meta-heuristic that imitates the process of natural selection and can be used to generate good solutions to search and optimization problems. GA has been successfully applied to a large number of problems and real-world applications such as timetabling scheduling, travelling salesman problem and its applications, vehicle routing problems, bioinformatics, scheduling applications, robotics, etc. The basics of a genetic algorithm was formulated by Holland J.P. in his fundamental paper [14], and were further developed by a number of other researchers such as Goldberg D.E. [13], among others.

In a GA, each individual represents a potential solution to a given problem. In a simple GA, each individual is a binary string. The string is called a chromosome. Each bit of the string is called a gene. The set of individuals, where each of them is a potential solution, is called a population. The population evolves (i.e. by serial conversion of one finite set of solutions to another by means of genetic operators: selection, crossover and mutation) in order to find an optimal or suboptimal solution. GA uses the mechanisms of natural evolution based on the following principles:

1. The first principle is based on the concept of survival of the fittest individual in natural selection according to Darwin[6], that states that the individuals that are more suited for solving problems in their environment are more likely to survive and reproduce more often. In a GA, by analogy with this principle, individuals with the better value of the objective (fitness) function are more likely to survive and reproduce.

2. The second principle is due to the fact that the chromosome of a child consists of the parts obtained from the parental chromosomes. This principle was established in 1865 by Mendel [19]. Its formalization provides a basis for the crossover operator.

3. The third principle is based on the concept of mutation - the significant changes in the characteristics of children and their acquisition of characteristics that are absent in their parents. By analogy with this principle, GA use a similar mechanism for the abrupt change in the characteristics of the children, and thus

increase the diversity (variability) of individuals in the population (the set of potential solutions).

These three principles form the core of the GA. Using them, the population evolves from generation to generation. The reproduction process produces a new set of individuals that replace the old population, and this process continues iteratively until a termination criterion is met, and the solution is returned.

2.2.1 Representation

In GA system a chromosome representation for the solution space is really critical in order to apply GA to a specific problem. The GA chromosome representation has to be chosen in the way to be expressive and powerful enough to represent a solution. Examples of chromosome representation include binary representation where a binary vector used to represent a chromosome (Figure 2.1).

1	0	0	0	1	0

Figure 2.1: Binary chromosome representation

Another example is order (permutation) representation (Figure 2.2).

1	3	2	4	6	5

Figure 2.2: Order-based chromosome representation

2.2.2 Population Initialization

There are many ways to create the initial population. The commonly used one involves randomly created individuals. Each created individual should represent a potential solution in the problem search space.

2.2.3 A GA Algorithm

In a GA, a population of chromosomes is maintained. Each chromosome is a potential solution to a problem. The chromosomes are altered or combined together by means of genetic operations, such as crossover and mutation. In crossover, two chromosomes are combined and two offspring are created and added to the new population. Mutation adds random properties to a chromosome.

The chromosomes are combined or altered repeatedly until a termination criterion is satisfied. In order to select chromosomes for crossover and mutation, a selection method is used. The same selection method may be used to inject existing individuals to the next generation. A fitness function is introduced in order to judge the quality of each potential solution. Usually, in GA, the termination criterion is when a certain number of generations has been reached and processed. Figure 2.3 and Figure 2.4 show the workflow and outline of a GA system.

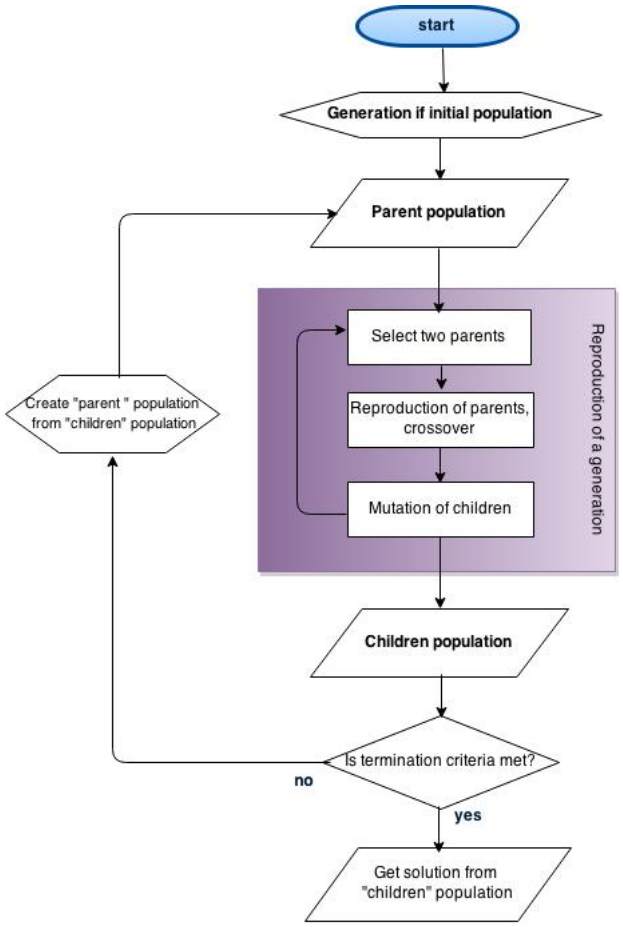

Figure 2.3: GA workflow

2.2.4 Fitness Evaluation

One of the most important concepts of GA is a fitness function. Fitness function determines how well the program is able to solve a problem and provides the information about program efficiency. Ideally, the function converts each chromosome to a potential solution to the given problem and then assigns a quality score to it. Finding a "correct" fitness function is very important in order to obtain a reasonable result.

```
Read problem instance data
Set GA parameters
Initial population, population_size is generated
for gen := 1 to maxGen {
    Fitness Evaluation population_size individuals
    Select new population
    Apply GA operators (crossover and mutation)
};
```

Figure 2.4: An outline of the genetic system

Usually, there are two kinds of problems based on a fitness function - maximization and minimization. In a maximization problem a chromosome with the higher fitness value is considered as the best solution. In a minimization problem a chromosome with the lower fitness value is considered as the best solution. The choice of a fitness function is dependent on the problem's objective.

2.2.5 Recombination

Recombination in GA usually takes the form of crossover where two chromosomes, selected during the selection step, undergo reproduction to produce offspring by exchanging genes between them.

2.2.6 Mutation

Mutation in GA is a minor alternation, in most cases random, of genes in the provided chromosome. In order to perform mutation we need only one chromosome at the time, as a result we get a mutated chromosome which is used in further generations. Mutation provides a diversity in the solutions and prevents them from convergence to a local optima.

2.2.7 Elitism

The idea of elitism is to allow the best individual(s) (chromosome(s)) from the current generation to carry over to the next generation unaltered during the construction of a new population. It guarantees that the solution quality will not decrease from one generation to the next.

2.2.8 Fitness Based Selection

Selection is a stage of GA where a chromosome is selected from a population for future breeding and recombination and/or mutation. The selection process is usually based on a quality of a chromosome, known as fitness. In fitness based selection an individual with better quality (fitness value) has bigger chance to be selected for recombination and mutation. Different probabilistic selection methods were developed in order to select individuals for reproduction. These include tournament selection, roulette wheel, rank selection and so on.

2.3 Multi-Objective Optimization

Optimization is a process of comparing solutions and finding the best one among some set of feasible solutions. Solutions can be good or bad depending on some objective, which could be a price, quality of a product, efficiency of a process, etc.. Apart from solving single goals, many real world problems have several objectives. One example of a multi-objective problem could be buying a car, where a person is trying to decrease the cost, and increase the quality and comfort of the car. Another example is a factory production process, when the production cost should be minimized and the quality of the product maximized. These situations refer to the area of multi-objective optimization [5, 8] (also called multi-criteria or multi-attribute optimization), the goal of which is to discover a number of trade-off optimal solutions respecting all the conflicting objectives of the problem.

Depending on the problem, there are some fitness evaluation strategies, helping to determine the general fitness of an individual. We now take a closer look at three of evaluation strategies weighted sum, Pareto ranking and normalized sum of ranks.

2.3.1 Weighted Sum

Weighted sum takes a multi-objective problem and converts it to a single-objective by adding weight to each parameter of each objective and combining them to a single objective function. This function is called a weighted sum.

$$Fitness = f_1 * W_1 + f_2 * W_2 + . + f_n * W_n \tag{2.1}$$

According to this formula, one score is appointed to the individual, and this situation can be viewed as a single-objective problem. We may change the importance of each

objective by changing its weight.

This evaluation strategy is limited by the facts that discovering suitable weights takes much time, and sometimes the identification of more important objective is biased.

2.3.2 Pareto Ranking

This type of evaluation strategy was named after Italian economist Vilfredo Pareto. It suggests using the dominant concept to score ranks for every individual in a population. At the same time, raw fitness values are substituted with the ranks. Pareto ranking [13] is successfully used in multi-objective optimization. Pareto ranking approach threats several objectives individually and thus eliminates the need for prior knowledge of weights, so it treats the problem as multi-objective rather than transforming it to a single objective problem.

X dominates Y:

$$\forall i : X_i \leq Y_i \wedge \exists i : X_i < Y_i \tag{2.2}$$

Following the strategy for the ranking of individuals, at first we go through the population, find non-dominated individuals and assign a rank of 1 as their score. After this, the ranked set of individuals is removed from the population. Next, we search for non-dominated individuals in the remaining unranked population and assign them a rank of 2. They are also removed from the unranked population.

This process continues until the entire population is assigned with a rank, and the given rank is considered to be a fitness score of the particular individual.

	Row scores			
Individual	Objective 1	Objective 2	Objective 3	Pareto rank
1	500	34	18	**1**
2	389	23	35	**2**
3	421	45	19	**2**
4	300	13	21	**1**
5	351	41	15	**1**

Figure 2.5: Pareto ranking example

Method:
currentrank = 1
N = popsize
m = n
while *m ≥ 1* **do**
  for *i=1 to m* **do**
    if *v_i is non-dominated* **then**
      rank(v_i) = currentrank
    end
  end
  for *j=1 to m* **do**
    if *rank(v_i) == currentrank* **then**
      remove v_j from population
      N = N-1
    end
  end
  currentrank = currentrank+1
  m=N
end

Algorithm 1: Pareto Ranking Scheme Algorithm [23]

2.3.3 Sum of Ranks

Sum of ranks [3] is another fitness evaluation strategy used in many multi-objective problems. In most cases it is applied for multi-objective problems with numerous dimensions. A great advantage of this method is the absence of outliers in solutions. This approach works in the following way: individuals who have the best fitness score of a given objective are assigned by rank 1; the next groups of best individuals are appointed with rank 2 and so forth until every individual is ranked. After ranking of all the objectives the fitness score of the individual is formed by calculating the sum of all ranks, using the formula:

$$fitness = \sum_{i=1}^{k} r_i * W_i \tag{2.3}$$

There are two variants of this approach. The first of them is explained above overall score is resulting from the sum of raw ranks. The second variant works by dividing each of the ranks of a certain objective by the sum of all its ranks. It is called the normalized sum of ranks. It is based on the principle of fair rank distribution

throughout the entire population.

After that, raw fitness of the individuals is replaced with the received summed ranks, and an appropriate reproduction selection strategy is employed.

Individual	Objective 1		Objective 2		Objective 3		Sum of ranks	Normalized ranks
	raw score	rank	raw score	rank	raw score	rank		
1	500	5	34	3	18	2	10	0.67
2	389	3	23	2	35	5	10	0.67
3	421	4	45	5	19	3	12	0.80
4	300	1	13	1	21	4	6	0.40
5	351	2	41	4	15	1	7	0.47
Sum		15		15		15		

Figure 2.6: Sum of rank scoring example

2.4 Information Systems

A data set is represented as a table, consisting of objects (rows) and attributes (columns). Each row represents an object, which could be an event, patient, human, animal, etc.. Every column represents some attribute, which could be measured for the object. It could be some observation, variable, property, measures, etc.. This table is called an information system (Table 2.1). An information table (system) [15] is a pair $A = (U,A)$, where U is a non-empty finite set of objects, A is a non-empty finite set of attributes, such that $a : U \rightarrow V_a$ for every a in A, where V_a is a set of attribute values of a.

As it was mentioned in [25], the operationalisation assumes the "nominal scale restriction" that

- Each object has exactly one value for each attribute at a given point in time;
- The observation of this value has no error.

2.5 Decision System

In many applications the outcome of classification is known. These tables have one distinguished attribute providing some a posteriori knowledge. This attribute is called a decision attribute. Information systems of this kind are called decision systems (Table 2.2).

A decision system $\mathcal{D}$ in the sense of Pawlak [24] is a tuple $\langle U, A, d \rangle$, where U is a finite set of objects and A is a finite set of attributes or features. Each $a \in A$ is

	Price	Guarantee (months)	Screen (inches)	Memory (Gb)	OS	Camera (megapixels)
x_1	low	6	small	16	Android	5
x_2	medium	6	small	32	iOS	5
x_3	medium	6	large	8	Windows	5
x_4	high	6	large	16	Android	5
x_5	high	12	small	8	Android	5
x_6	low	18	large	8	Windows	5
x_7	low	24	small	32	Android	5
x_8	medium	18	small	32	iOS	5
x_9	high	12	small	8	Windows	5

Table 2.1: Information System example

considered as a function with domain U and some range V_a. The attribute $d \notin A$ is the decision attribute.

	Price	Guarantee (months)	Screen (inches)	Memory (Gb)	OS	Camera (megapixels)	**Demand**
x_1	low	6	small	16	Android	5	**high**
x_2	medium	6	small	32	iOS	5	**low**
x_3	medium	6	large	8	Windows	5	**low**
x_4	high	6	large	16	Android	5	**low**
x_5	high	12	small	8	Android	5	**low**
x_6	low	18	large	8	Windows	5	**low**
x_7	low	24	small	32	Android	5	**high**
x_8	medium	18	small	32	iOS	5	**high**
x_9	high	12	small	8	Windows	5	**high**

Table 2.2: Decision Table example

Many real-life datasets may be unnecessarily large because some of the attributes may be superfluous for making decisions. In the Table 2.3 it clear that the "Camera" is superfluous and does not have any influence on the decision attribute.

In case of the dataset above we can get rid of the superfluous attribute, which we do not need for future computations.

2.6 Rough Sets

Rough set theory was first described by the Polish computer scientist Zdzisaw Pawlak in the early 1980s [24] and has become a widely used research tool for knowledge

	Price	Guarantee (months)	Screen (inches)	Memory (Gb)	OS	Camera (megapixels)	**Demand**
x_1	low	6	small	16	Android	5	**high**
x_2	medium	6	small	32	iOS	5	**low**
x_3	medium	6	large	8	Windows	5	**low**
x_4	high	6	large	16	Android	5	**low**
x_5	high	12	small	8	Android	5	**low**
x_6	low	18	large	8	Windows	5	**low**
x_7	low	24	small	32	Android	5	**high**
x_8	medium	18	small	32	iOS	5	**high**
x_9	high	12	small	8	Windows	5	**high**

Table 2.3: Decision Table example with the superfluous attribute coloured in grey

discovery. As it was mentioned in [21], a Rough set is a formal approximation of a crisp set such as a conventional set, in terms of a pair of sets which gives the lower and the upper approximation of the original set.

The basic assumption of Rough Set Data Analysis is that information is presented and perceived up to a certain granularity [9]

The philosophy of Rough Sets [9] is the following:

- Knowledge has a granular structure.
  - Granularity can be expressed by the classes of some equivalence relation.
- Due to granularity, distinct objects can be indiscernible.
- The extension of a concept can only be approximated.

Rough sets have real-life applications in many different spheres like economics, management, engineering, etc.

In [18], a hybrid approach to bankruptcy classification using Genetic Programming(GP) and Rough Sets was introduced. In this paper authors investigated a hybrid approach to bankruptcy prediction using a genetic programming to construct a bankruptcy prediction model with variables from the RS model. The reducts are used as an input for GP, where GP is supposed to construct a prediction model. According to the paper, their model was 80% accurate on a validation sample compared to original RS model which was 67% accurate.

In [12], the author showed how RS can support the design of lean workflow systems, particularly lean decision rules at decision gate by detecting excessive and redundant information as well as how RS can be utilized to disclose missing information in

a workflow system. Also, the authors demonstrated how RS can help predict the routing of a process within a workflow system.

In [26], an example of RS application in a forecasting module of natural hazards' monitoring system in hard-coal mines was presented. RSs were applied to reduction a number of variables used during the hazard assessment, checking the variables' significance and defining decision rules.

2.6.1 Indiscernibility

Each attribute set B determines an equivalence relation θ_B on U. This relation called indiscernibility relation or B-indiscernibility relation.

Definition 1. *$x \equiv_{\theta_B} y$ if and only if $a(x) = a(y)$ for all $a \in B$.*

It means that x and y can not be distinguished with the attributes in B.

	Screen	OS	Demand
x_1	large	Windows	low
x_2	large	iOS	high
x_3	large	Android	high
x_4	small	Windows	low
x_5	small	iOS	low
x_6	small	Android	high
x_7	small	iOS	high
x_8	small	Android	low

Table 2.4: Decision Table. Indiscernibility

The indiscernibility classes of the table 2.4 are shown below:
$\theta_{\{Screen\}} = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6, x_7, x_8\}\}$
$\theta_{\{OS\}} = \{\{x_1, x_4\}, \{x_2, x_5, x_7\}, \{x_3, x_6, x_8\}\}$
$\theta_{\{Screen,OS\}} = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5, x_7\}, \{x_6, x_8\}\}$
$\theta_{\{Demand\}} = \{\{x_1, x_4, x_5, x_8\}, \{x_2, x_3, x_6, x_7\}\}$
With each class K of θ_B we can associate a feature vector t^K in such a way that all elements of K take the same value in t^K.

2.6.2 Approximation

As it was mentioned before, a rough set is a formal approximation of a crisp set in terms of a pair of sets which gives the lower and the upper approximation of the original set. Now it is time to define the lower and the upper approximation.

The B-lower approximation is denoted as $\underline{B}X$ and has the following formula:

$$\underline{B}X = \{x \in U | \theta_B(x) \subseteq X\} \tag{2.4}$$

The B-upper approximation is denoted as $\overline{B}X$ and has the following formula:

$$\overline{B}X = \{x \in U | \theta_B(x) \cap X \neq \emptyset\} \tag{2.5}$$

The B-boundary region of X is a difference between upper approximation and lower approximation

$$BN_B(X) = \overline{B}X \setminus \underline{B}X \tag{2.6}$$

To get a better idea about the approximation and its application the example below is provided.

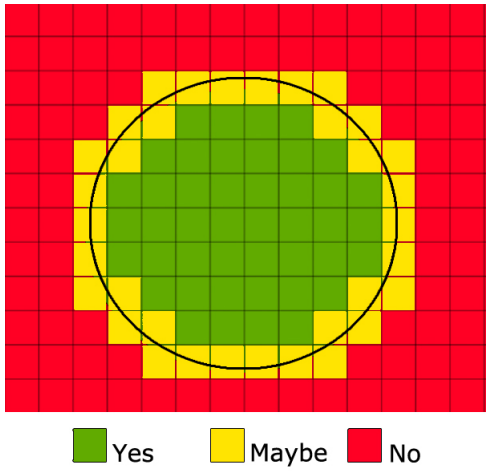

Figure 2.7: Upper and lower approximation

Let X = $\{x : Demand(x) = high\}$ (the circle in Figure 2.7),
$B = \{Screen, OS\}$ (Table 2.5)

The indiscernibility classes are:
$\theta_B = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5, x_7\}, \{x_6, x_8\}\}$
For this table we have the following approximations and boundary-region:
$\underline{B}X = \{x_2, x_3\}$
$\overline{B}X = \{x_2, x_3, x_5, x_6, x_7, x_8\}$
$BN_B(X) = \{x_5, x_6, x_7, x_8\}$

	Screen	OS	Demand
x_1	large	Windows	low
x_2	large	iOS	high
x_3	large	Android	high
x_4	small	Windows	low
x_5	small	iOS	low
x_6	small	Android	high
x_7	small	iOS	high
x_8	small	Android	low

Table 2.5: Decision Table

2.6.3 Dependency

We are interested in finding rules for some attribute set $B \subseteq A$ of the form

$$(\forall x \in U)(\forall a \in B)[a(x) = t_a \text{ implies } d(x) = s] \tag{2.7}$$

where t_a is a feature vector with attributes restricted to B, and s is in the domain of the decision attribute.

It means that the value of each x on d is uniquely determined by its values on B. Property 2.7 could be rewritten with algebraic equivalent of it:

$$\theta_B \subseteq \theta_d \tag{2.8}$$

It is called a functional dependency and denoted as $B \Rrightarrow_U d$.

The decision attribute d functionally depends on the attribute set B with respect to all objects in U if and only if any objects in U have the same attributes in B, then the decision attributes d of these objects must be the equal.

2.6.4 Reducts

Refer to property 2.7, a reduct for d is a non-empty subset B of A such that $B \Rrightarrow_U d$ and for which if $C \subsetneq B$, then $C \not\Rrightarrow d$.

In other words, an attribute set $B \subseteq A$ is called a reduct if B is a minimal with respect to $\theta_B \subseteq \theta_d$

In traditional rough set theory, feature selection via reducts is an important area.

In [9, 15, 21, 25, 29, 35], theoretical properties of reducts were introduced and deeply analyzed. The importance of reducts was shown in the works above.

In [29], authors investigated the notion of an association reduct. They mentioned that the association reducts represent data-based functional dependencies between the sets of attributes, where it is preferred that possibly smallest sets determine possibly largest sets. Also they compared the notions of an association reduct to other types of reducts previously studied within the theory of rough sets.

In [21, 25, 29, 31], reduct calculation methods using discernibility matrices and Boolean reasoning were introduced.

Another method of calculating reduct is using genetic algorithms. Despite the fact that this method was proposed and discussed in several works [4, 25], we did not find enough details on the GA implementation part, analysis of an influence of different genetic operators and fitness evaluation methods on the results.

There could be a several number of reducts for some decision table. Analysing the decision table shown before (Table 2.2), the reducts are {Guarantee,Screen,OS} and {Price,Guarantee,OS}.

2.6.5 Rules

If $B \Rrightarrow_U d$, and the classes of θ_B are $X_1, ..., X_n$ with associated feature vectors $t_1, ...t_n$, then for each i there is some equivalence class Y_j with value s_j such that $X_i \subseteq Y_j$. This defines one deterministic rule:

$$(\forall x \in U, a \in B)[a(x) = t_i \text{ implies } d(x) = s_j]$$

There is one such deterministic rule for each class X_i of θ_B.

Analyzing Table 2.2 we obtain the reduct {Price,Guarantee,OS}.

	Price	Guarantee (months)	Screen (inches)	Memory (Gb)	OS	Camera (megapixels)	**Demand**
x_1	low	6	small	16	Android	5	**high**
x_2	medium	6	small	32	iOS	5	**low**
x_3	medium	6	large	8	Windows	5	**low**
x_4	high	6	large	16	Android	5	**low**
x_5	high	12	small	8	Android	5	**low**
x_6	low	18	large	8	Windows	5	**low**
x_7	low	24	small	32	Android	5	**high**
x_8	medium	18	small	32	iOS	5	**high**
x_9	high	12	small	8	Windows	5	**high**

Table 2.6: Decision reduct

For the decision table above (Table 2.6), taking into account the reduct, we can find the following deterministic rules: (Price, low) and (Guarantee, 6 months) ⇒ (Demand, high)
(Price, medium) and (Guarantee, 6 months) ⇒ (Demand, low)
(Price, medium) and (Guarantee, 18 months) ⇒ (Demand, high)
(Guarantee, 24 months) and (OS, Android) ⇒ (Demand, high)
(Guarantee, 18 months) and (OS, Windows) ⇒ (Demand, low)
(Guarantee, 12 months) and (OS, Windows) ⇒ (Demand, high)

If $B \not\Rightarrow_U d$, and the classes of θ_B are $X_1, ..., X_n$ with associated feature vectors $t_1, ...t_n$ then there are equivalence classes of θ_d $Y_0, ..., Y_k$, $k > 0$, with values $s_0, ..., s_k$ such that for some i $X_i \cap Y_0 \neq \emptyset, \ldots, X_i \cap Y_k \neq \emptyset$. This defines one indeterministic rule:

$$(\forall x \in U, a \in B)[a(x) = t_i \text{ implies } d(x) = s_0 \text{ or... or } d(x) = s_k]$$

For Table 2.5 we can define the rules below.

Deterministic rules:
(OS, Windows) ⇒ (Demand, low)
(Screen, large) and (OS, iOS) ⇒ (Demand, high)
(Screen, large) and (OS, Android) ⇒ (Demand, high)

Indeterministic rules:
(Screen, small) ⇒ (Demand, low) or (Demand, high)
(OS, iOS) ⇒ (Demand, low) or (Demand, high)
(OS, Android) ⇒ (Demand, low) or (Demand, high)

2.6.6 Bireducts

As finding reducts is computationally unfeasible [27], many methods of finding feature sets with an acceptable quality of classification were proposed within classical rough set theory; see, for example, [2, 20, 28].

In [30] it was mentioned that the original definition of a reduct is restrictive, requiring that it should determine decisions or, if data inconsistencies do not allow full determinism, provide the same level of information about decisions as the complete set of attributes. As well as the idea of using approximate reducts [29, 31] - that

almost preserve the decision information - has some notable issues.

Recently, given the challenges above, Ślęzak and Janusz [30] proposed to take into account not only the horizontal reduction of information by feature selection, but also a vertical reduction considering suitable subsets of the original set of objects. Thus, the aim is to find areas in the two-dimensional object $\times$ attribute plane which are in some sense best suited for classification. This leads to the following definition: A *(decision) bireduct* is a pair $\langle B, X\rangle$ such that $B \subseteq A, X \subseteq U$ and

R1. For all $b \in B, x, y \in X$, $b(x) = b(y)$ implies $d(x) = d(y)$; in this case, we write $B \Rrightarrow_X d$. (B is discerning all elements of X)

R2. If $C \subsetneq B$, there are $x, y \in X$ such that $c(x) = c(y)$ for all $c \in C$ and $d(x) \neq d(y)$. (Minimality of B with respect to X)

R3. If $X \subsetneq Y$, there are $x, y \in Y$ such that $b(x) = b(y)$ for all $b \in B$ and $d(x) \neq d(y)$. (Maximality of X with respect to B)

Because we optimize the decision table in two-dimensions, decreasing the number of attributes and increasing the number of objects these attributes are valid for, there could be a huge number of bireducts for any given decision table.

Analyzing Table 2.2 we obtain a big number of bireducts.

One of the bireducts is ({Guarantee,OS},{1-3,5-9}) (Table 2.7).

	Price	Guarantee (months)	Screen (inches)	Memory (Gb)	OS	Camera (megapixels)	**Demand**
x_1	low	6	small	16	Android	5	**high**
x_2	medium	6	small	32	iOS	5	**low**
x_3	medium	6	large	8	Windows	5	**low**
x_4	high	6	large	16	Android	5	**low**
x_5	high	12	small	8	Android	5	**low**
x_6	low	18	large	8	Windows	5	**low**
x_7	low	24	small	32	Android	5	**high**
x_8	medium	18	small	32	iOS	5	**high**
x_9	high	12	small	8	Windows	5	**high**

Table 2.7: Decision bireduct example 1. ({Guarantee,OS},{1-3,5-9})

Another bireduct is ({Price},{1-5}) (Table 2.8).

Few more examples of decision bireducts:

({Guarantee,Screen,OS},{1-9})

({Price,Guarantee,OS},{1-9})

	Price	Guarantee (months)	Screen (inches)	Memory (Gb)	OS	Camera (megapixels)	**Demand**
x_1	low	6	small	16	Android	5	**high**
x_2	medium	6	small	32	iOS	5	**low**
x_3	medium	6	large	8	Windows	5	**low**
x_4	high	6	large	16	Android	5	**low**
x_5	high	12	small	8	Android	5	**low**
x_6	low	18	large	8	Windows	5	**low**
x_7	low	24	small	32	Android	5	**high**
x_8	medium	18	small	32	iOS	5	**high**
x_9	high	12	small	8	Windows	5	**high**

Table 2.8: Decision bireduct example 2. ({Price},{1-5})

({Price,Guarantee},{1-4,6-9})
({Screen},{1,3-4,6-9})
({Price,OS},{1-7,9})
({OS},{1-3,6-7})

Decision rules now can be obtained as in classical rough set theory by restricting the scope of quantifiers to the parts of a bireduct $\langle B, X\rangle$. Bireducts – which are consistent on their object set – may be viewed as inducing approximate rules on the whole object set U.

For Table 2.7 we can define the following deterministic rules:

(Guarantee, 6 months) and (OS, Android) ⇒ (Demand, high)
(Guarantee, 6 months) and (OS, iOS) ⇒ (Demand, low)
(Guarantee, 6 months) and (OS, Windows) ⇒ (Demand, low)
(Guarantee, 12 months) and (OS, Android) ⇒ (Demand, low)
(Guarantee, 18 months) and (OS, Windows) ⇒ (Demand, low)
(Guarantee, 24 months) and (OS, Android) ⇒ (Demand, high)
(Guarantee, 18 months) and (OS, iOS) ⇒ (Demand, high)
(Guarantee, 12 months) and (OS, Windows) ⇒ (Demand, high)

For Table 2.8 we can define the following deterministic rules:

(Price, low) ⇒ (Demand, high)
(Price, medium) ⇒ (Demand, low)
(Price, high) ⇒ (Demand, low)

Besides the introduction of the notion of a bireduct, in [30] the authors investigated the ability of ensembles of bireducts characterized by significant diversity with respect to both subset of attributes that describe decisions and a subset of objects, for which this subset of attributes is valid to represent the knowledge hidden in data and to serve as means for learning robust classification system. The authors used 3 datasets from UCI Machine Learning Repository [17] for their experiment: zoo , lymphography and SPECT. They proposed a random search algorithm in order to find bireducts with a different size of objects and attributes. The permutation of n+m numbers was used, where n is a number of objects and m is a number of attributes. One of the main reasons of using a random search, as it was noted by authors, is that even for relatively small data sets, it is impossible to examine all permutations.

First of all, they compared decision bireducts with decision reducts in term of their size. For each dataset, they computed 1000 decision reducts and 14000 bireducts for 14 different values of the ratio parameter. In the paper above, they investigate the ratios - the higher the number, the more attributes appear early in the sequence - spanning from from 0 to $2|U|/|A|$. As it was expected, the average number of objects covered by a bireduct went down when the ratio was increased.

Authors have also investigated the influence of the ratio parameter and classifier aggregation methods on the results of the classification. Two aggregation methods were used: the majority voting and the balanced support weighted voting [31]. The majority voting scheme classifies a test object to the decision class indicated by the highest number of triggered rules derived from bireducts in the ensemble [30]. The balanced support weighted voting scheme weights each vote using the support of the corresponding rule[30]. The results clearly showed the strengths of bireduct ensembles. As it was mentioned by authors, their scores on all three datasets are comparable with the best results in machine learning literature [10, 16]. They showed that in prediction bireducts performed better than reducts. Bireducts were able to recognize significantly more objects compared to reducts.

Following the work mentioned above, in our research we propose a new approach to generate bireducts using a multi-objective genetic algorithm. Although the GAs were used to calculate reduct in some previous works[4, 35], we did not find any work where GAs were adopted to calculate bireducts.

Chapter 3

System Architecture

This chapter provides the details of the system implementation by focusing on the GA and RS system components. The details of the proposed GA including chromosome representation, genetic operators and fitness evaluation strategies are provided here.

3.1 MOGA System

Slightly modifying the algorithm proposed in [30], we use a GA chromosome as an input, compared to the random search algorithm output used in [30]. We developed our own GA system to find bireducts as given in Figure 3.1. Firstly, the program reads the parameter file and provided dataset(s). Because many attributes in the datasets we use are numerical, the system performs discretization on these attribute values. Next, the GA system initializes the first population. During evaluation phase, the system finds bireduct for the dataset based on provided chromosome. After the system found a bireduct, it calculates fitness for the current chromosome based on the bireduct, taking into account the number of attributes and objects. In case of Pareto ranking and "normalized sum of ranks" the system ranks all individuals in the end of each generation.

Appendix C shows several examples of bireducts generated by proposed MOGA system.

3.1.1 Chromosome Representation and Initial Population Creation

An order-based chromosome representation was employed for encoding the chromosome. The representation scheme uses integers in the range of 1...n+m, where n is

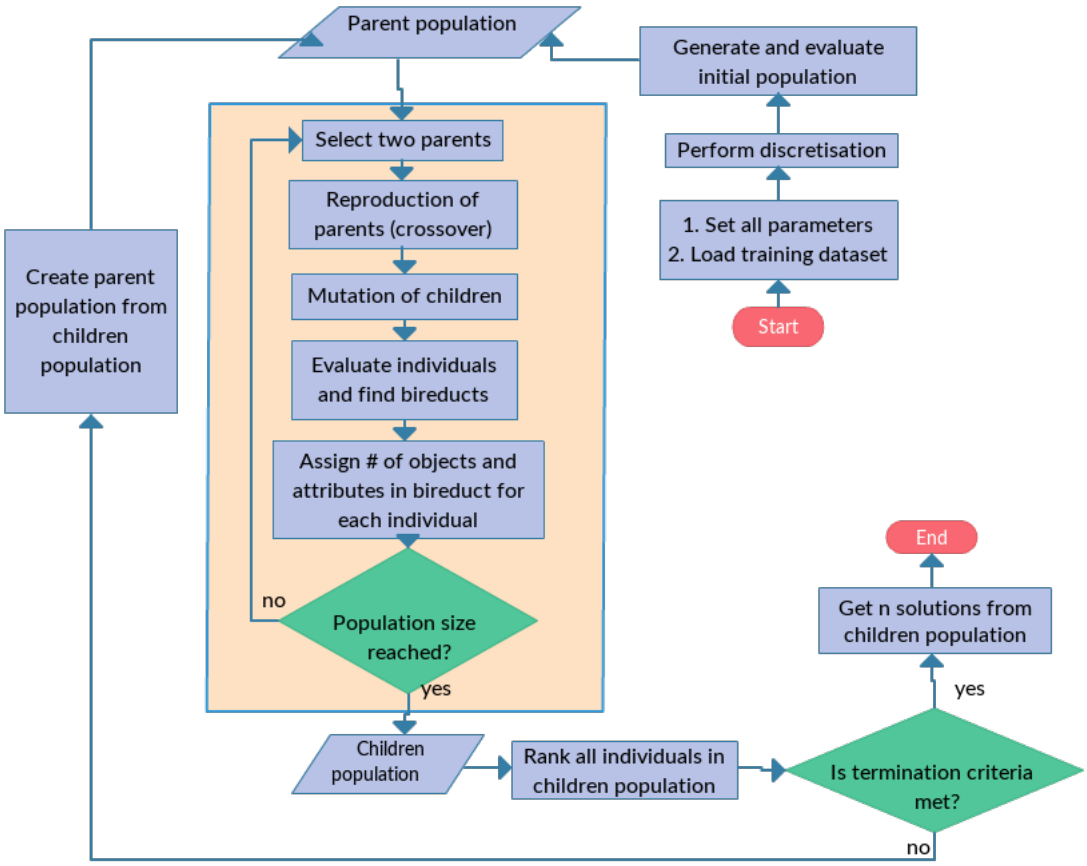

Figure 3.1: MOGA system workflow

a number of instances (objects) and m is a number of conditional attributes in the provided as input dataset.

The initial population is represented by a random permutation of the numbers in the range of 1...n+m.

3.1.2 Selection

The selection method was used in proposed MOGA system is a tournament selection. Tournament selection is a method of selecting an individual from a population of individuals in GA. In a tournament selection k (tournament size) individuals are chosen at random from the population. The winner of the "tournament" (the individual with the best fitness value) is selected for later reproduction. Since for crossover operation two individuals are needed, two tournament selections are made.

Tournament selection works in the following way:

- choose k (the tournament size) individuals from the population at random

- choose the best individual based on its fitness/rank from the tournament
- return the selected individual

Selection pressure is easily adjusted by changing the tournament size. If the tournament size is large, weak individuals have a smaller chance to be selected. If tournament size is too small, for example, is one, then the GA is reduced to a random search.

In case of Pareto ranking our system selects n (output size) random individuals with the lowest rank from the population in the last generation.

3.1.3 Fitness Evaluation Strategies

Weighted Sum

Detailed information about the weighted sum was shown in Section 2.3.1. The weighted sum approach transforms a multi-objective optimization problem into a single objective optimization problem by adding the problem objective functions together, (i.e., in this case, minimize the number of attributes and maximize the number of objects that attributes are valid for) by assigning weighted coefficients for each individual objective.

Finding the "right" weights is really important in order to get a reasonable result. The procedure of finding the suitable weights is not trivial. The weights used in our research were empirically established after several runs.

We want to compare two approaches, where in first one we assign a larger weight to the first objective(obj_1), which is to minimize the number of attributes. In the second approach we assign a larger weight to the second objective(obj_2). This is to maximize the number of instances the attributes are valid for, or in our case, minimize the difference between the number of instances in original dataset and number of instances received in bireduct.

$$f(x) = obj_1 * w_{obj_1} + obj_2 * w_{obj_2} \tag{3.1}$$

After applying equation 3.1 using different weight, it was empirically established to use $weight_{obj_1} = 1$, $weight_{obj_2} = 0.5$ for the first approach and $weight_{obj_1} = 0.7$, $weight_{obj_2} = 1.5$ for the second.

Pareto ranking

Detailed information about the Pareto ranking was shown in Section 2.3.2.

We have two objectives:

- Minimize the number of attributes.
- Maximize the number of objects that attributes are valid for.

To convert it to minimization problem we change the objectives a little bit, so the Pareto ranking algorithm was supplied with two values:

1. Number of attributes (obj_1).
2. Difference between number of objects in the training dataset and number of objects in the provided bireduct (obj_2).

Normalized Sum Of Ranks

The Sum Of Ranks is discussed in Section 2.3.3. The Sum Of Ranks algorithm was supplied with two values:

1. Number of attributes.
2. Difference between number of objects in the training dataset and number of objects in the provided bireduct.

The smaller the sum of the objectives, the better quality it is. Both objectives were treated equally, so no weights were assigned.

3.1.4 Crossover

Ordered Crossover

One of the crossover operators used in the experiments is the ordered crossover (OX) [7]. In this kind of crossover two random numbers (r_1,r_2) are generated in the beginning. Then genes from position in range of r_1 to r_2 of the first parent are copied to the corresponding gene positions of the first offspring. Then, starting from the position r_2, we copy all genes from parent 2 to offspring 1, avoiding repetition. The same operation is performed for the second parent and offspring. Figure 3.2 shows an example of ordered crossover.

	0	1	2	3	4	5	6	7	8
Parent 1 :	4	9	3	7	2	6	8	1	5
Parent 2 :	5	6	8	2	3	4	7	1	9
Offspring 1 :	5	3	4	7	2	6	8	1	9
Offspring 2 :	9	2	6	8	3	4	7	1	5

Figure 3.2: Ordered crossover example. r_1=4, r_2 =6.

Cycle Crossover

Another crossover operator used is cycle crossover (CX) [22]. CX creates offspring from ordered chromosome by identifying so-called cycles between two provided parent chromosomes. The cycles are then copied from the respective parent chromosomes in order to create children. We used CX with random start where the gene to start with is identified by random number. CX with the random start works in the following way:

1. Start from the random generated position (r) of the first parent.
2. Go the the gene at the same position of the second parent.
3. Look at the position with the same gene in the first parent.
4. Add the gene index to the cycle.
5. Repeat steps 2-5 till we get the the first gene of the cycle.

In order the generate children, the indices of the cycle are used in alternating order. For example, genes if parent 1 are copied to offspring 1 in cycle 1, genes of parent 1 copied to offspring 2 in cycle 2, and so on...

Figure 3.3 shows an example of cycle crossover.

	0	1	2	3	4	5	6	7	8
Parent 1 :	4	9	3	7	2	6	8	1	5
Parent 2 :	5	6	8	2	3	4	7	1	9
Offspring 1 :	4	9	8	2	3	6	7	1	5
Offspring 2 :	5	6	3	7	2	4	8	1	9

Figure 3.3: Cycle crossover example. r = 2.

3.1.5 Mutation

Reciprocal Exchange

In our experiments two different mutation operators were used. Reciprocal exchange is one of them. In reciprocal exchange mutation we choose randomly gene in the chromosome (n_1), then we choose randomly another gene in the same chromosome(n_2), after that we exchange two selected genes.

An example of reciprocal exchange for chromosome 4-9-3-7-2-6-8-1-5 is provided in Figure 3.4.

	0	1	2	3	4	5	6	7	8
Chromosome before mutation :	4	9	3	7	2	6	8	1	5
Chromosome after mutation :	4	9	3	7	8	6	2	1	5

Figure 3.4: Reciprocal Exchange mutation example. $n_1 = 4$, $n_2 = 6$.

Inversion

Another mutation operator was used is inversion. In Figure 3.5 the example is shown.

Inversion mutation works in the following way:

1. Select 2 random positions in chromosome, start position p_1 and end position p_2.
2. Inverse the genes in the p_1 - p_2 interval.
3. Select random insertion position in, insert the genes received after step 2 after the insertion point.

	0	1	2	3	4	5	6	7	8
Chromosome before mutation	4	9	3	7	2	6	8	1	5
Step 1	4	9	3	7	2	6	8	1	5
Step 2	4	9	3	8	6	2	7	1	5
Step 3	4	9	8	6	2	7	3	1	5
Chromosome after mutation	4	9	8	6	2	7	3	1	5

Figure 3.5: Inversion mutation example. p_1=3, $p_2 = 6$, $in = 1$.

3.1.6 Elitism

An elitism size of 1 was used. The best individual was always copied into the next generation.

3.2 RS System

After our MOGA system generates all bireducts for the provided dataset, we pass the bireducts to our RS system which generates decision rules based on bireducts and applies it to the testing dataset. The workflow of the proposed MOGA sustem is shown in Figure 3.6.

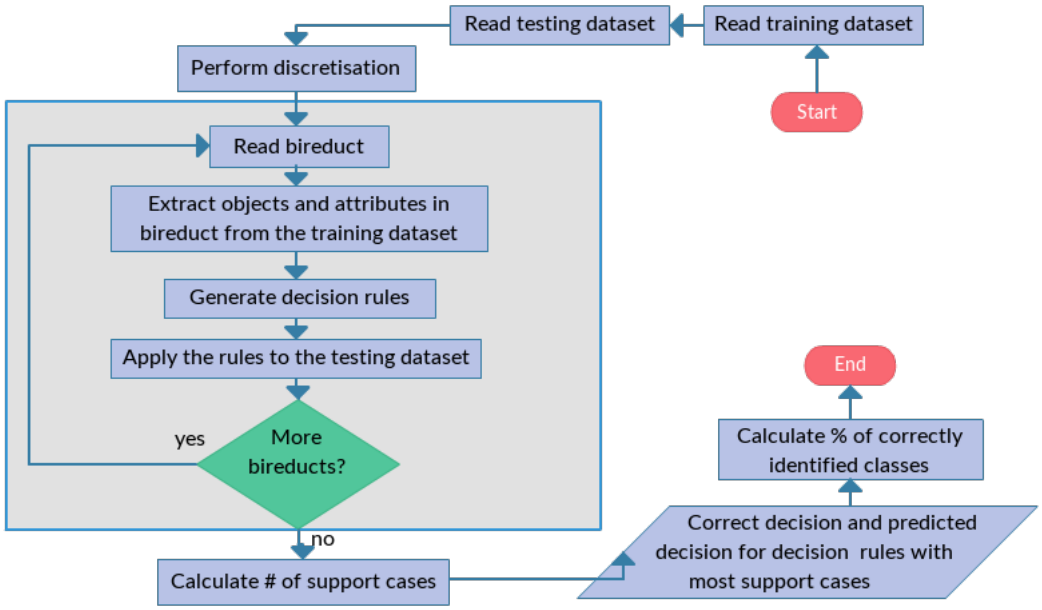

Figure 3.6: RS system workflow

Chapter 4

Experiments

This chapter provides the datasets used, analysis and discussion of the results of the performed experiments. Accuracy comparison of the performance of the different fitness evaluation strategies and genetic operators was carried out.

4.1 Benchmark Datasets

To perform our experiments we used three datasets from UCI Machine Learning Repository [17]. The datasets are SPECT Heart, Breast Cancer Wisconsin (Diagnostic) and Spambase. These data sets are explained below.

Name	SPECT Heart	Breast Cancer Wisconsin (Diagnostic)	Spambase	Ionosphere	Pima
Instances	267	569	4601	351	768
Attributes	22	32	57	34	8
Training	234	512	4026	328	692
Testing	33	57	575	23	76

Table 4.1: Benchmark Datasets

4.1.1 SPECT Heart Data Set

The dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. The dataset contains observations of 267 patients with one SPECT image per patient. Each patient is classified into one of the two categories: normal or abnormal. The SPECT Heart Data Set from UCI Machine Learning Repository [17] has 267 instances with 22 attributes. All the attributes are binary.

Cross-validation technique was employed where 234 instances were used for training and 33 instances for testing.

4.1.2 Breast Cancer Wisconsin (Diagnostic)

The dataset describes the features that are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The features describe characteristics of the cell nuclei present in the image. Each observation is classified into one of the two categories: malignant or benign. The Breast Cancer Wisconsin (Diagnostic) dataset from UCI Machine Learning Repository [17] has 569 instances with 32 attributes. Cross-validation technique was employed where 512 instances were used for training and 57 instances for testing.

4.1.3 Spambase Data Set

The Spambase Data Set from UCI Machine Learning Repository [17] is a collection of spam e-mails which came from postmaster and individuals who had filed spam. Each observation is classified into one of the two categories: considered spam (1) or not (0). The dataset has 4601 instances with 57 attributes. Cross-validation technique was employed where 4026 instances were used for training and 575 instances for testing.

4.1.4 Ionosphere Data Set

The Ionosphere Data Set from UCI Machine Learning Repository [17] is a collection of a radar data by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. Each observation is classified into one of the two categories: “good” or “bad”. The dataset has 351 instances with 34 attributes. Cross-validation technique was employed where 328 instances were used for training and 23 instances for testing.

4.1.5 Pima Indians Diabetes Data Set

The Pima Indians Diabetes Data Set from UCI Machine Learning Repository [17] is a collection of data from patients who are females at least 21 years old of Pima Indian heritage. Each observation is classified into one of the two categories: diagnosis positive (1) or diagnosis negative (0). The dataset has 768 instances with 8 attributes. Cross-validation technique was employed where 692 instances were used for training and 76 instances for testing.

4.2 Experimental Set-up

The proposed MOGA system was implemented in Java and all experiments were performed on Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz with 4GB RAM on an Ubuntu 15.04 environment. Several experiments were performed for the datasets listed in Section 4.1. For each experiment 30 runs were made.

The experiment performed were:

1. *Accuracy comparison using different fitness evaluation methods and genetic operators.* 30 runs for 2 datasets from UCI Machine Learning Repository [17]: Breast Cancer Wisconsin (Diagnostic) and Spambase were made using different fitness evaluation strategies and genetic operators:

    - Weighted sum, OX and RE
    - Weighted sum, CX and RE
    - Weighted sum, OX and Inversion
    - Weighted sum, CX and Inversion
    - Normalized sum of ranks, OX and RE
    - Normalized sum of ranks, CX and RE
    - Normalized sum of ranks, OX and Inversion
    - Normalized sum of ranks, CX and Inversion
    - Pareto ranking, OX and RE
    - Pareto ranking, CX and RE
    - Pareto ranking, OX and Inversion
    - Pareto ranking, CX and Inversion

2. *Accuracy comparison of proposed method with other known results.* Average prediction accuracy of 30 runs for 5 datasets from UCI Machine Learning Repository [17]: Breast Cancer Wisconsin (Diagnostic), Spambase, SPECT Heart, Ionosphere and Pima Indians Diabetes Datasets is compared with results of other known methods.

Table 4.2 gives the GA parameters used for different datasets. Ordered (OX) and cycle (CX) crossover operators, reciprocal exchange (RE) and inversion mutation operators were used. Also different fitness evaluation strategies like weighted sum, Pareto ranking and "normalized sum of ranks" were used.

In our version of GA system n best individuals (chromosomes in output) are saved in the result file (output) after the termination criterion reached.

Parameter	WDBC	Spambase	SPECT	Ionosphere	Pima
Population size	500	450	300	200	200
Generation span	50	30	30	30	30
Selection type	Tournament	Tournament	Tournament	Tournament	Tournament
Tournament size	4	4	3	3	2
Crossover operator	OX, CX	OX, CX	OX	OX	OX
Crossover rate	0.9	0.9	0.9	0.9	0.9
Mutation operator	Inversion, RE	Inversion, RE	RE	RE	Inversion
Mutation rate	0.1	0.1	0.1	0.25	0.3
Elitism	1	1	1	1	1
Fitness evaluation strategy	Weighted sum, normalized sum of ranks, Pareto ranking	Weighted sum, normalized sum of ranks, Pareto ranking	Pareto ranking	Normalized sum of ranks	Normalized sum of ranks
Chromosomes in output	300	400	150	150	150
Number of GA runs	30	30	30	30	30

Table 4.2: GA parameters.

4.3 Accuracy comparison using different fitness evaluation methods and genetic operators.

In the subsections below the comparison of results for two datasets using different genetic operators and fitness evaluation methods are shown.

4.3.1 Weighted sum

OX and RE

The average prediction accuracy overall 30 runs for WDBC dataset using ordered crossover, reciprocal exchange mutation and weighted sum fitness evaluation method with $weight_{obj_1} = 0.7$, $weight_{obj_2} = 1.5$ is 93.90%. On average, only 54 out of 300 generated bireducts were unique and used to generate rules.

The graph in Figure 4.1 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around generation 25 it is eventually approach the best fitness. After that it was observed that best and average fitness are distancing, meaning a higher diversity within the population. Figure A.1 shows that the MOGA System was able to significantly

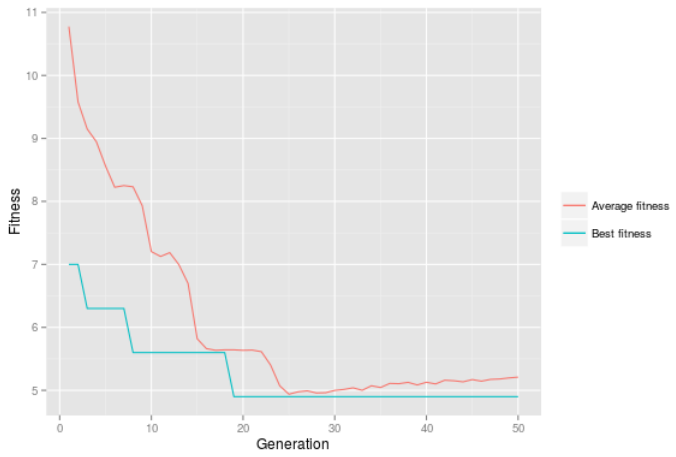

Figure 4.1: Fitness vs generations for WDBC dataset (average of 30 runs). Weighted Sum (0.7,1.5), OX and RE .

reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 475 objects and 10 attributes.

The average prediction accuracy overall 30 runs for Spam dataset with $weight_{obj_1}$ = 0.7, $weight_{obj_2}$ = 1.5 is 85.53%. On average, only 9 out of 300 generated bireducts were unique and used to generate rules.

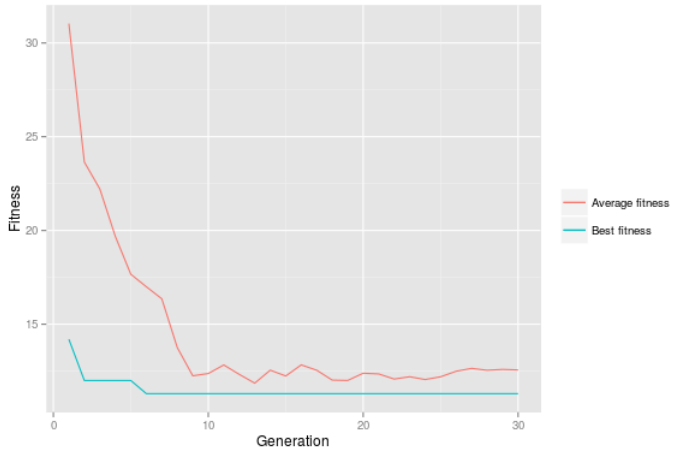

Figure 4.2: Fitness vs generations for Spam dataset (average of 30 runs). Weighted Sum (0.7,1.5), OX and RE.

The graph in Figure 4.2 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value dramatically improved till generation 9. After that the average fitness stays around the same value but with small variations.

Figure A.2 shows that the MOGA system was able to significantly reduce the

number of attributes without much reduction of the number of objects. The average size of bireducts received is 4025 objects and 15 attributes.

The average prediction accuracy overall 30 runs for WDBC dataset with $weight_{obj_1}$ = 1, $weight_{obj_2}$ = 0.5 is 90.92%. On average, only 36 out of 300 generated bireducts were unique and used to generate rules.

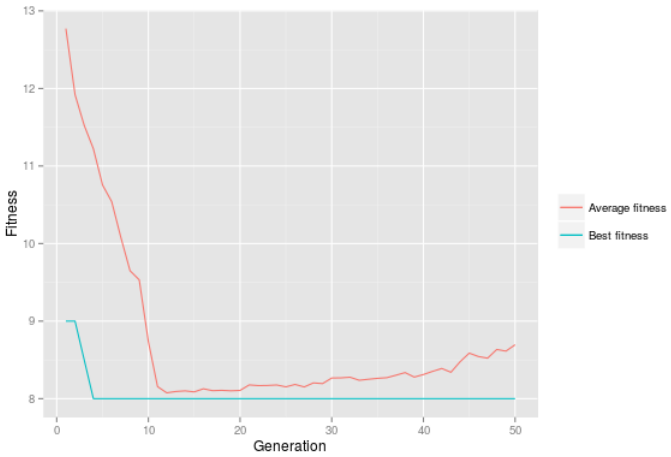

Figure 4.3: Fitness vs generations for WDBC dataset (average of 30 runs). Weighted Sum (1, 0.5), OX and RE.

The graph in Figure 4.3 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around generation 11 it is eventually almost approach the best fitness. After that it was observed that best and average fitness are distancing, meaning a higher diversity within the population.

Figure A.3 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 475 objects and 9 attributes.

The average prediction accuracy overall 30 runs for Spam dataset with $weight_{obj_1}$ = 1, $weight_{obj_2}$ = 0.5 is 85.20%. On average, only 6 out of 400 generated bireducts were unique and used to generate rules.

The graph in Figure 4.4 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around generation 16 it is eventually approach the best fitness. After that it was observed that best and average fitness are distancing, meaning a higher diversity within the population.

Figure A.4 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average

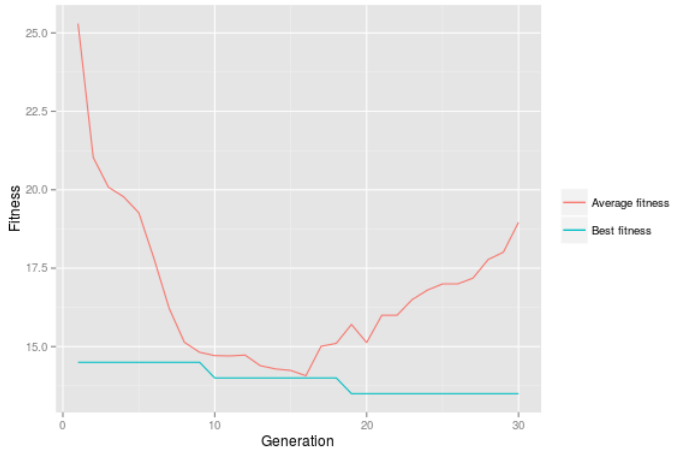

Figure 4.4: Fitness vs generations for Spam dataset (average of 30 runs). Weighted Sum (1, 0.5), OX and RE.

size of bireducts received is 4024 objects and 13 attributes.

CX and RE

The average prediction accuracy overall 30 runs for WDBC dataset with $weight_{obj_1}$ = 1, $weight_{obj_2}$ = 0.5 is 87.19%. On average, 66 out of 300 generated bireducts were unique and used to generate rules.

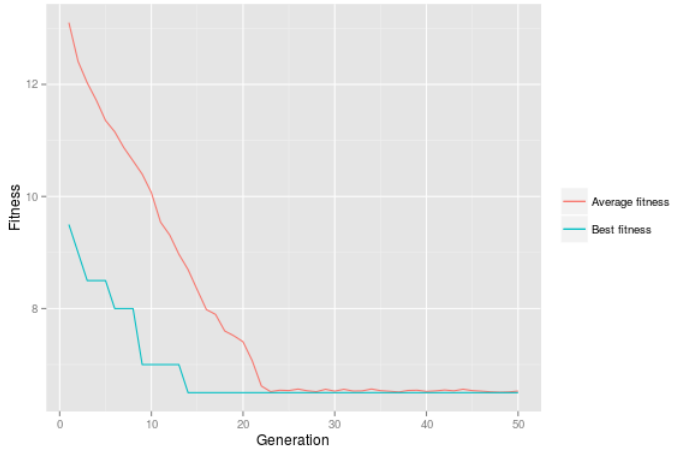

Figure 4.5: Fitness vs generations for WDBC dataset (average of 30 runs). Weighted Sum (1, 0.5), CX and RE.

The graph in Figure 4.5 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around generation 22 it is eventually approach the best fitness. After that we can see

that average and best fitness plots stocked together, which means that the population has converged.

Figure A.5 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 475 objects and 8 attributes.

The average prediction accuracy overall 30 runs for Spam dataset with $weight_{obj_1}$ = 1, $weight_{obj_2}$ = 0.5 is 84.77%. On average, 326 out of 400 generated bireducts were unique and used to generate rules.

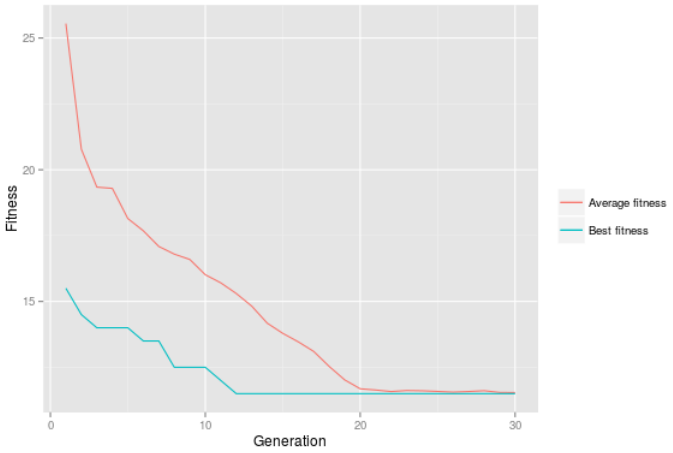

Figure 4.6: Fitness vs generations for Spam dataset (average of 30 runs). Weighted Sum (1, 0.5), CX and RE.

The graph in Figure 4.6 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around generation 21 it is eventually approach the best fitness. After that we can see that average and best fitness plots stocked together, which means that the population has converged.

Figure A.6 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4024 objects and 14 attributes.

The average prediction accuracy overall 30 runs for WDBC dataset with $weight_{obj_1}$ = 0.7, $weight_{obj_2}$ = 1.5 is 93.90%. On average, 101 out of 300 generated bireducts were unique and used to generate rules.

The graph in Figure 4.7 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around generation 19 it is eventually approach the best fitness. After that we can see that average and best fitness plots stocked together, which means that the population

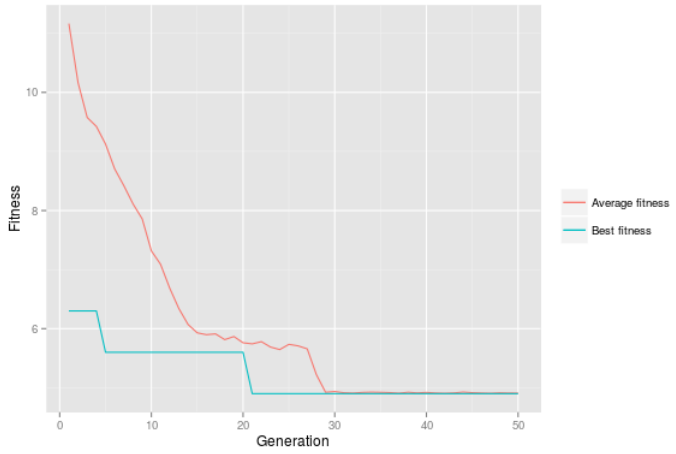

Figure 4.7: Fitness vs generations for WDBC dataset (average of 30 runs). Weighted Sum (0.7, 1.5), CX and RE.

has converged.

Figure A.7 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 475 objects and 9 attributes.

The average prediction accuracy overall 30 runs for Spam dataset with $weight_{obj_1}$ = 0.7, $weight_{obj_2}$ = 1.5 is 84.13%. On average, only 13 out of 400 generated bireducts were unique and used to generate rules.

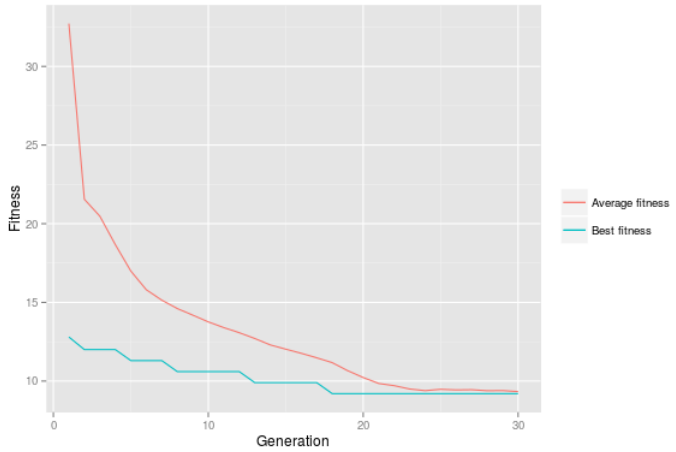

Figure 4.8: Fitness vs generations for Spam dataset (average of 30 runs). Weighted Sum (0.7, 1.5), CX and RE.

The graph in Figure 4.8 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around generation 22 it is eventually approach the best fitness.

Figure A.8 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4025 objects and 12 attributes.

OX and Inversion

The average prediction accuracy overall 30 runs for WDBC dataset with $weight_{obj_1}$ = 1, $weight_{obj_2}$ = 0.5 is 89.14%. On average, only 25 out of 300 generated bireducts were unique and used to generate rules.

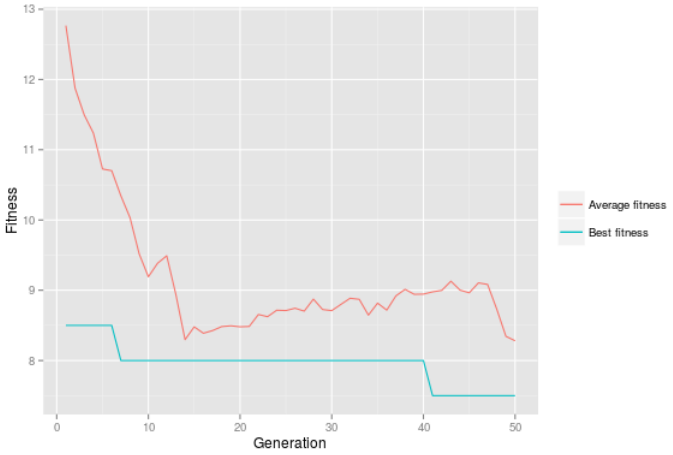

Figure 4.9: Fitness vs generations for WDBC dataset (average of 30 runs). Weighted Sum (1, 0.5), OX and Inversion.

The graph in Figure 4.9 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around the 14th generation it is eventually get close to the best fitness.

Figure A.9 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 475 objects and 9 attributes.

The average prediction accuracy overall 30 runs for Spam dataset with $weight_{obj_1}$ = 1, $weight_{obj_2}$ = 0.5 is 87.53%. On average, only 4 out of 400 generated bireducts were unique and used to generate rules.

The graph in Figure 4.10 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value dramatically improved till 13th generation. After that the average fitness stays around the same value but with small variations.

Figure A.10 shows that the MOGA system was able to significantly reduce the

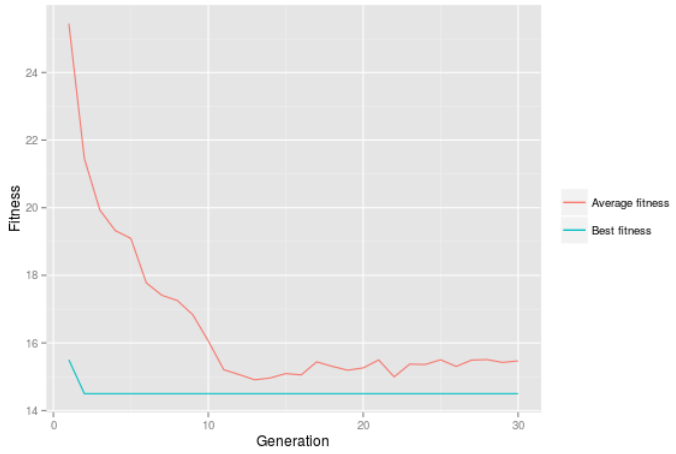

Figure 4.10: Fitness vs generations for Spam dataset (average of 30 runs). Weighted Sum (1, 0.5), OX and Inversion.

number of attributes without much reduction of the number of objects. The average size of bireducts received is 4024 objects and 14 attributes.

The average prediction accuracy overall 30 runs for WDBC dataset with $weight_{obj_1}$ = 0.7, $weight_{obj_2}$ = 1.5 is 92.73%. On average, only 14 out of 300 generated bireducts were unique and used to generate rules.

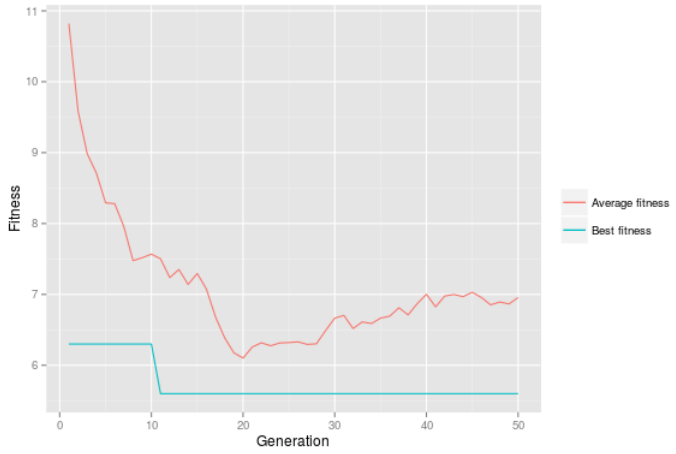

Figure 4.11: Fitness vs generations for WDBC dataset (average of 30 runs). Weighted Sum (0.7, 1.5), OX and Inversion.

The graph in Figure 4.11 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around 20th generation it is eventually get close to the best fitness, but after that it has started distancing

Figure A.11 shows that the MOGA system was able to significantly reduce the

number of attributes without much reduction of the number of objects. The average size of bireducts received is 475 objects and 9 attributes.

The average prediction accuracy overall 30 runs for Spam dataset with $weight_{obj_1}$ = 0.7, $weight_{obj_2}$ = 1.5 is 87.29%.On average, only 14 out of 400 generated bireducts were unique and used to generate rules.

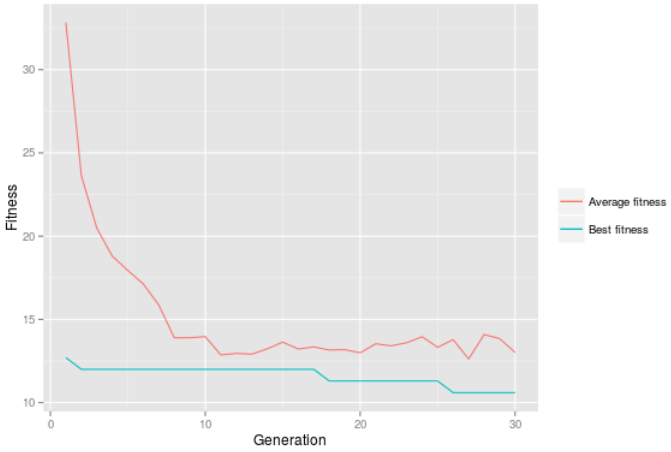

Figure 4.12: Fitness vs generations for Spam dataset (average of 30 runs). Weighted Sum (0.7, 1.5), OX and Inversion.

The graph in Figure 4.12 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value dramatically improved till 11th generation. After that the average fitness stays around the same value but with small variations.

Figure A.12 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4023 objects and 16 attributes.

CX and Inversion

The average prediction accuracy overall 30 runs for WDBC dataset with $weight_{obj_1}$ = 0.7, $weight_{obj_2}$ = 1.5 is 89.78%. On average, 69 out of 300 generated bireducts were unique and used to generate rules.

The graph in Figure 4.13 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value dramatically improved till 19th generation. After that the average fitness stays around the same value but with small variations.

Figure A.13 shows that the MOGA system was able to significantly reduce the

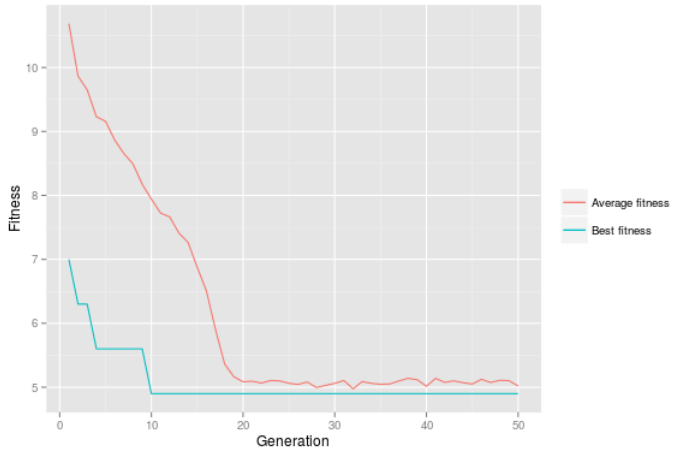

Figure 4.13: Fitness vs generations for WDBC dataset (average of 30 runs). Weighted Sum (0.7, 1.5), CX and Inversion.

number of attributes without much reduction of the number of objects. The average size of bireducts received is 475 objects and 9 attributes.

The average prediction accuracy overall 30 runs for Spam dataset using Cycled crossover, inversion mutation and weighted sum fitness evaluation method with $weight_{obj_1} = 0.7$, $weight_{obj_2} = 1.5$ is 83.66%. On average, 41 out of 400 generated bireducts were unique and used to generate rules.

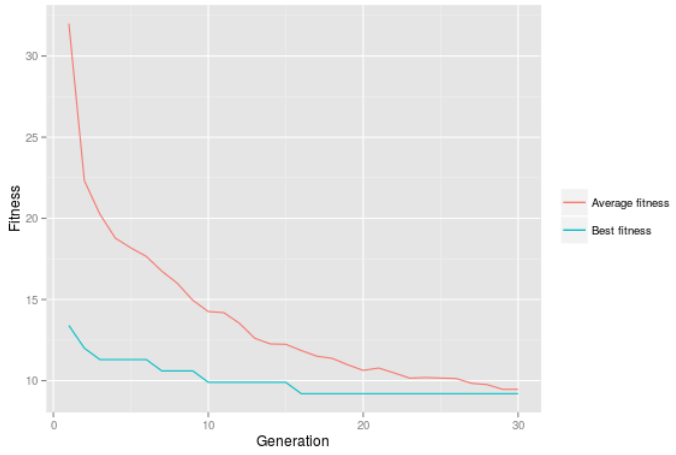

Figure 4.14: Fitness vs generations for Spam dataset (average of 30 runs). Weighted Sum (0.7, 1.5), CX and Inversion.

The graph in Figure 4.14 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value improves and at around 30th generation it is eventually approach the best fitness.

Figure A.14 shows that the MOGA system was able to significantly reduce the

number of attributes without much reduction of the number of objects. The average size of bireducts received is 4025 objects and 13 attributes.

The average prediction accuracy overall 30 runs for WDBC dataset with $weight_{obj_1}$ = 1, $weight_{obj_2}$ = 0.5 is 88.86%. On average, 40 out of 300 generated bireducts were unique and used to generate rules.

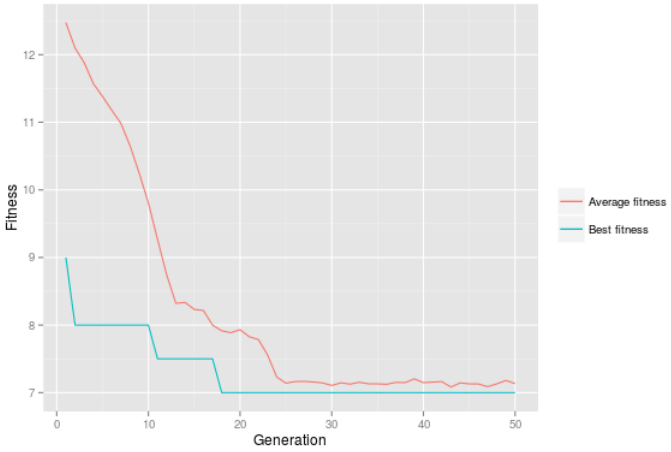

Figure 4.15: Fitness vs generations for WDBC dataset (average of 30 runs). Weighted Sum (1, 0.5), CX and Inversion.

The graph in Figure 4.15 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value dramatically improved till 25th generation. After that the average fitness stays around the same value but with small variations.

Figure A.15 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 475 objects and 8 attributes.

The average prediction accuracy overall 30 runs for Spam dataset with $weight_{obj_1}$ = 1, $weight_{obj_2}$ = 0.5 is 83.34%. On average, 334 out of 400 generated bireducts were unique and used to generate rules.

Figure A.16 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4022 objects and 14 attributes.

The graph in Figure 4.16 shows that the training performed well and as evolution progresses the average fitness that started from a bad fitness value eventually approach the best fitness in the end of the run.

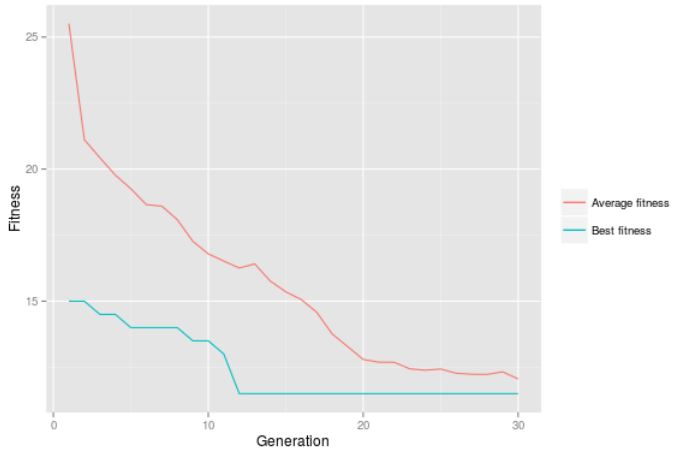

Figure 4.16: Fitness vs generations for Spam dataset (average of 30 runs). Weighted Sum (1, 0.5), CX and Inversion.

Analysis

Table 4.3 shows the prediction accuracies, number if bireducts used and the sizes of bireducts for weighted sum fitness evaluation method and WDBC dataset.

Parameters	Min accuracy	Max accuracy	Average accuracy	Average # of bireducts used (out of 300)	Average size of bireduct (o x a)
Weighted Sum(0.7,1.5), CX and RE	78.72	100	93.90	101	475 x 9
Weighted Sum(1, 0.5), CX and RE	61.70	100	87.19	66	475 x 8
Weighted Sum(0.7,1.5), OX and RE	70.21	100	93.90	54	475 x 10
Weighted Sum(1, 0.5), OX and RE	55.31	100	90.92	36	475 x 9
Weighted Sum(0.7,1.5), CX and Inversion	64.89	98.93	89.78	69	475 x 9
Weighted Sum(1, 0.5), CX and Inversion	61.70	100	88.86	40	475 x 8
Weighted Sum(0.7,1.5), OX and Inversion	65.95	100	92.73	14	475 x 9
Weighted Sum(1, 0.5), OX and Inversion	60.63	100	89.14	25	475 x 9

Table 4.3: Prediction Accuracy Comparison. Weighted Sum. WDBC dataset

Table 4.3 shows that different crossover and mutation operators do not signif-

icantly affect the prediction accuracy for WDBC dataset when the weighted sum fitness evaluation method is used. At the same time it was noticed that the ordered crossover brings higher diversity in the population. Also the weights of different objectives in the fitness function do not significantly affect the predictions accuracies. One-way ANOVA test results with the significance level of 0.05 for WDBC dataset are shown in TableB.1. The results showed that there are no significant difference between prediction accuracies when different genetic operators used.

Parameters	Min accuracy	Max accuracy	Average accuracy	Average # of bireducts used (out of 400)	Average size of bireduct (o x a)
Weighted Sum(0.7,1.5), CX and RE	82.95	85.21	84.13	13	4025 x 12
Weighted Sum(1, 0.5), CX and RE	83.47	85.73	84.77	326	4024 x 14
Weighted Sum(0.7,1.5), OX and RE	84.34	86.60	85.53	9	4025 x 15
Weighted Sum(1, 0.5), OX and RE	84.17	86.26	85.20	6	4024 x 13
Weighted Sum(0.7,1.5), CX and Inversion	82.60	84.69	83.66	41	4025 x 13
Weighted Sum(1, 0.5), CX and Inversion	82.43	84.69	83.34	334	4022 x 14
Weighted Sum(0.7,1.5), OX and Inversion	86.08	88.17	87.29	14	4023 x 16
Weighted Sum(1, 0.5), OX and Inversion	86.43	88.69	87.53	4	4024 x 14

Table 4.4: Prediction Accuracy Comparison. Weighted Sum. Spam dataset

Analysing the performance of our MOGA system during the trainings using weighted sum fitness evaluation method, we observed that the OX provides higher diversity in population compared to CX , and gives better prediction accuracies. Table 4.4 shows the prediction accuracies, number of bireducts used and the sizes of bireducts for weighted sum fitness evaluation method and Spam dataset. It was observed that the weights of different objectives in the fitness function do not significantly affect the predictions accuracies. One-way ANOVA test results with the significance level of 0.05 for WDBC dataset are shown in TableB.2. The results showed that the difference is significant between prediction accuracies in most cases when different genetic operators used.

4.3.2 Normalized Sum Of Ranks

OX and RE

The average prediction accuracy overall 30 runs for WDBC dataset is 97.44%. On average, 300 out of 300 generated bireducts were unique and used to generate rules.

Figure A.17 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 474 objects and 12 attributes.

The average prediction accuracy overall 30 runs for Spam dataset is 83.56%. On average, 399 out of 400 generated bireducts were unique and used to generate rules.

Figure A.18 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4017 objects and 18 attributes.

CX and RE

The average prediction accuracy overall 30 runs for WDBC dataset is 97.58%. On average, 300 out of 300 generated bireducts were unique and used to generate rules.

Figure A.19 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 474 objects and 12 attributes.

The average prediction accuracy overall 30 runs for Spam dataset is 83.77%. On average, 400 out of 400 generated bireducts were unique and used to generate rules.

Figure A.20 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4017 objects and 18 attributes.

OX and Inversion

The average prediction accuracy overall 30 runs for WDBC dataset is 97.51%. On average, 300 out of 300 generated bireducts were unique and used to generate rules.

Figure A.21 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 474 objects and 12 attributes.

The average prediction accuracy overall 30 runs for Spam dataset is 84.04%. On average, 400 out of 400 generated bireducts were unique and used to generate rules.

Figure A.22 shows that the MOGA system was able to significantly reduce the

number of attributes without much reduction of the number of objects. The average size of bireducts received is 4018 objects and 18 attributes.

CX and Inversion

The average prediction accuracy overall 30 runs for WDBC dataset is 97.37%. On average, 300 out of 300 generated bireducts were unique and used to generate rules.

Figure A.23 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 474 objects and 12 attributes.

The average prediction accuracy overall 30 runs for Spam dataset is 83.58%. On average, 400 out of 400 generated bireducts were unique and used to generate rules.

Figure A.24 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4018 objects and 18 attributes.

Analysis

Parameters	Min accuracy	Max accuracy	Average accuracy	Average # of bireducts used (out of 300)	Average size of bireduct (o x a)
CX and RE	95.74	98.93	97.58	300	474 x 12
OX and RE	94.68	98.93	97.44	300	474 x 12
CX and Inversion	94.68	98.93	97.37	300	474 x 12
OX and Inversion	94.68	98.93	97.51	300	474 x 12

Table 4.5: Prediction Accuracy Comparison. Normalized Sum Of Ranks. WDBC dataset

Table 4.5 shows the prediction accuracies, number of bireducts used and the sizes of bireducts for "normalized sum of ranks" fitness evaluation method and WDBC dataset. It was noticed that different crossover and mutation operators do not significantly affect the prediction accuracy for WDBC dataset. It was observed that the weights of different objectives in the fitness function do not significantly affect the predictions accuracies. One-way ANOVA test results with the significance level of 0.05 for WDBC dataset are shown in TableB.3. The results showed that there are no significant difference between prediction accuracies when different genetic operators used.

Parameters	Min accuracy	Max accuracy	Average accuracy	Average # of bireducts used (out of 400)	Average size of bireduct (o x a)
CX and RE	82.60	84.86	83.77	400	4017 x 18
OX and RE	82.43	84.69	83.56	399	4017 x 18
CX and Inversion	82.43	84.69	83.58	400	4018 x 18
OX and Inversion	82.78	85.04	84.04	400	4018 x 18

Table 4.6: Prediction Accuracy Comparison. Normalized Sum Of Ranks. Spam dataset

Table 4.6 shows the prediction accuracies, number of bireducts used and the sizes of bireducts for "normalized sum of ranks" fitness evaluation method and Spam dataset. OX and Inversion gave us the best prediction accuracy using normalized sum of ranks. One-way ANOVA test results with the significance level of 0.05 for WDBC dataset are shown in TableB.4. It showed that OX and Inversion gave significantly better prediction accuracy compared to OX and RE , but the difference is not significant compared to others.

4.3.3 Pareto ranking

OX and RE

The average prediction accuracy overall 30 runs for WDBC dataset is 97.44%. On average, 300 out of 300 generated bireducts were unique and used to generate rules.

Figure 4.17 the average number of rank 1 individuals in each generation.

Figure A.25 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 474 objects and 12 attributes.

The average prediction accuracy overall 30 runs for Spam dataset is 83.04%. On average, 400 out of 400 generated bireducts were unique and used to generate rules.

Figure 4.18 the average number of rank 1 individuals in each generation.

Figure A.26 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4017 objects and 18 attributes.

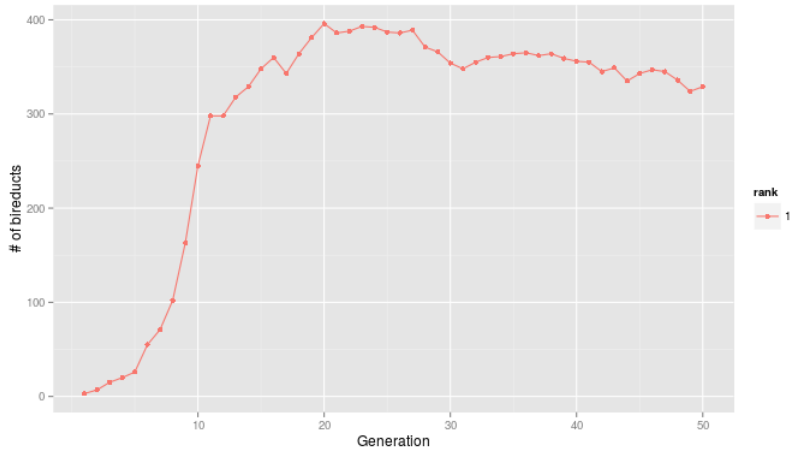

Figure 4.17: Number of rank 1 chromosomes vs generations for WDBC dataset (average of 30 runs). Pareto ranking, OX and RE.

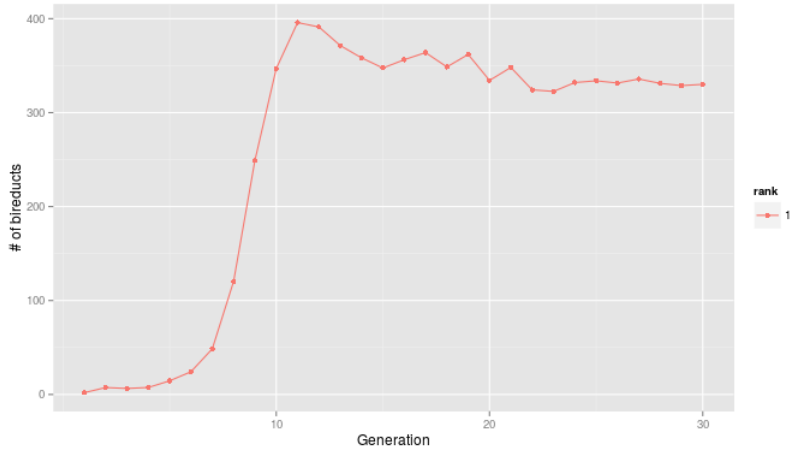

Figure 4.18: Number of rank 1 chromosomes vs generations for Spam dataset (average of 30 runs). Pareto ranking, OX and RE.

CX and RE

The average prediction accuracy overall 30 runs for WDBC dataset is 97.58%. On average, 300 out of 300 generated bireducts were unique and used to generate rules.

Figure 4.19 the average number of rank 1 individuals in each generation.

Figure A.27 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 474 objects and 12 attributes.

The average prediction accuracy overall 30 runs for Spam dataset is 83.29%. On average, 400 out of 400 generated bireducts were unique and used to generate rules.

Figure 4.20 the average number of rank 1 individuals in each generation.

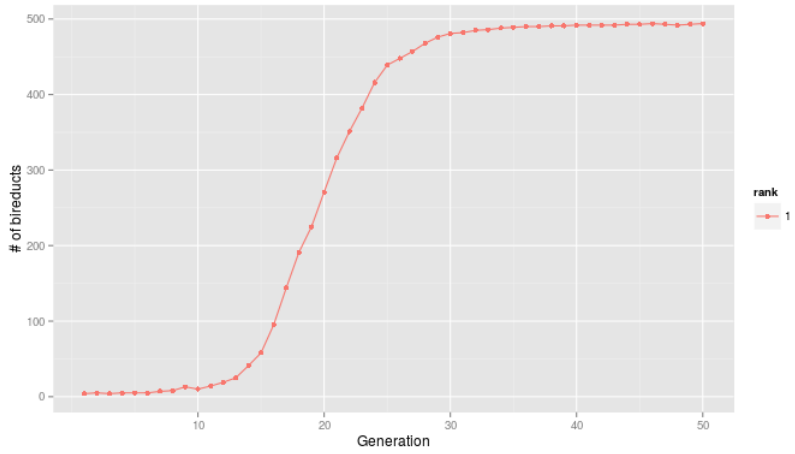

Figure 4.19: Number of rank 1 chromosomes vs generations for WDBC dataset (average of 30 runs). Pareto ranking, CX and RE.

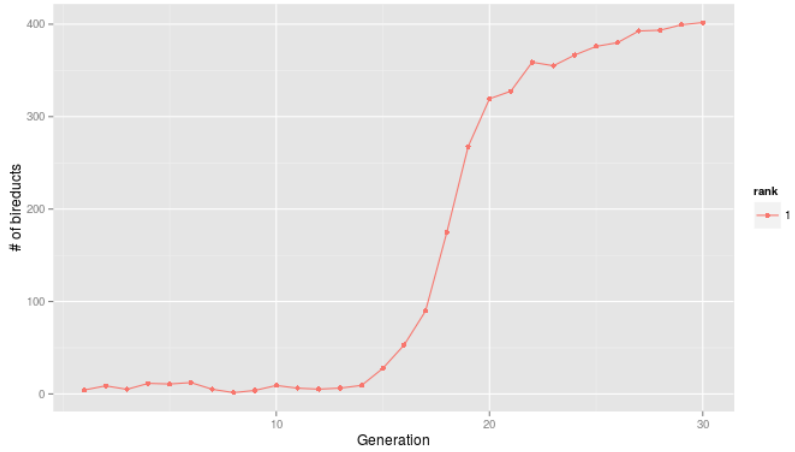

Figure 4.20: Number of rank 1 chromosomes vs generations for Spam dataset (average of 30 runs). Pareto ranking, CX and RE.

Figure A.28 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4018 objects and 18 attributes.

OX and Inversion

The average prediction accuracy overall 30 runs for WDBC dataset is 97.62%. On average, 300 out of 300 generated bireducts were unique and used to generate rules.

Figure 4.21 the average number of rank 1 individuals in each generation.

Figure A.29 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average

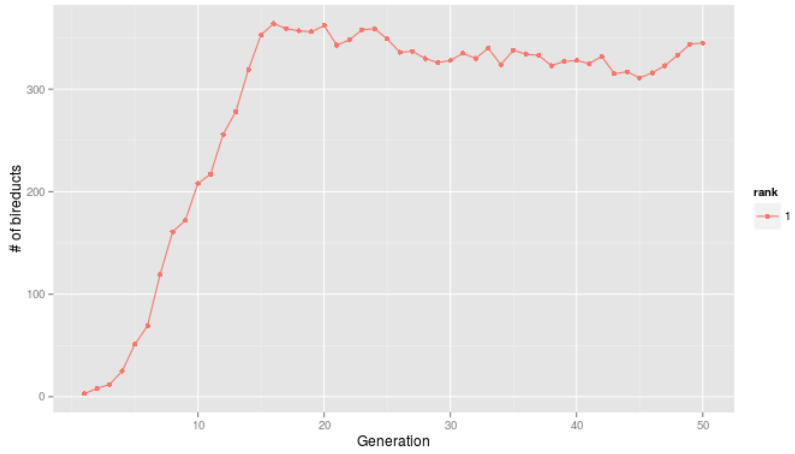

Figure 4.21: Number of rank 1 chromosomes vs generations for WDBC dataset (average of 30 runs). Pareto ranking, OX and Inversion.

size of bireducts received is 474 objects and 12 attributes.

The average prediction accuracy overall 30 runs for Spam dataset is 84.77%. On average, 400 out of 400 generated bireducts were unique and used to generate rules.

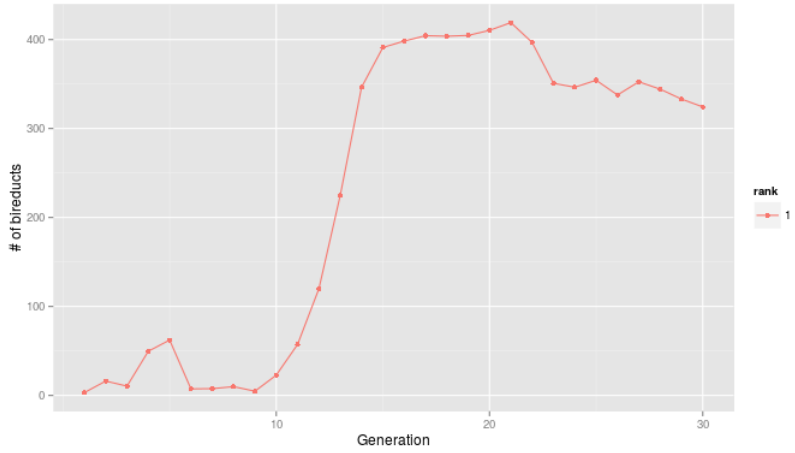

Figure 4.22: Number of rank 1 chromosomes vs generations for Spam dataset (average of 30 runs). Pareto ranking, OX and Inversion.

Figure 4.22 the average number of rank 1 individuals in each generation. Figure A.30 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average size of bireducts received is 4018 objects and 18 attributes.

CX and Inversion

The average prediction accuracy overall 30 runs for WDBC dataset is 97.62%. On average, 300 out of 300 generated bireducts were unique and used to generate rules.

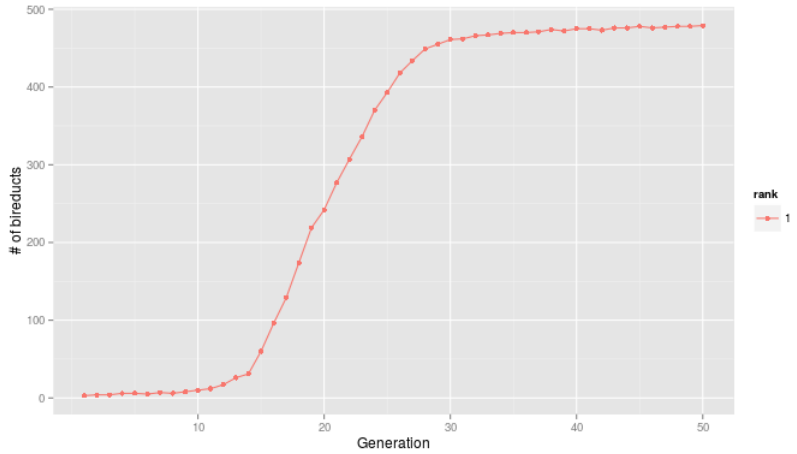

Figure 4.23: Number of rank 1 chromosomes vs generations for WDBC dataset (average of 30 runs). Pareto ranking, CX and Inversion.

Figure 4.23 the average number of rank 1 individuals in each generation.

The average prediction accuracy overall 30 runs for Spam dataset is 83.70%. On average, 400 out of 400 generated bireducts were unique and used to generate rules.

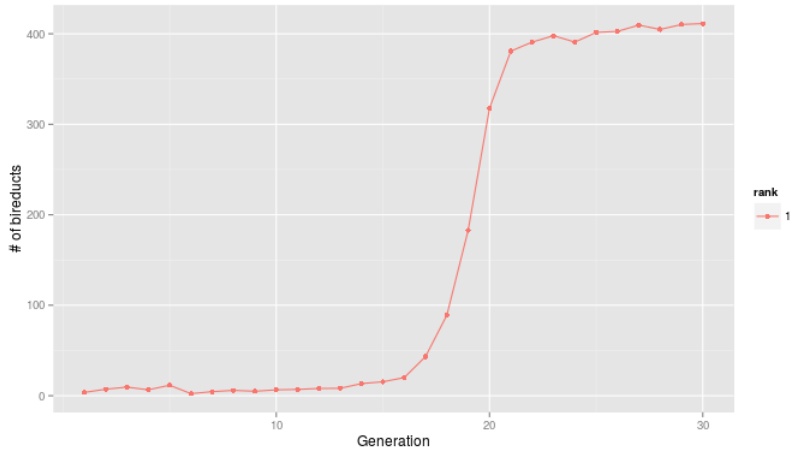

Figure 4.24: Number of rank 1 chromosomes vs generations for Spam dataset (average of 30 runs). Pareto ranking, CX and Inversion.

Figure 4.24 the average number of rank 1 individuals in each generation.

Figure A.32 shows that the MOGA system was able to significantly reduce the number of attributes without much reduction of the number of objects. The average

size of bireducts received is 4017 objects and 18 attributes.

Analysis

Parameters	Min accuracy	Max accuracy	Average accuracy	Average # of bireducts used (out of 300)	Average size of bireduct (o x a)
CX and RE	95.74	98.93	97.58	300	474 x 12
OX and RE	95.74	98.93	97.44	300	474 x 12
CX and Inversion	95.74	98.93	97.62	300	474 x 12
OX and Inversion	95.74	98.93	97.62	300	474 x 12

Table 4.7: Prediction Accuracy Comparison. Pareto ranking. WDBC dataset

Table 4.7 shows the prediction accuracies, number of bireducts used and the sizes of bireducts for Pareto ranking fitness evaluation method and WDBC dataset. It was noticed that different crossover and mutation operators do not significantly affect the prediction accuracy for WDBC dataset. It was observed that the weights of different objectives in the fitness function do not significantly affect the predictions accuracies. One-way ANOVA test results with the significance level of 0.05 for WDBC dataset are shown in TableB.5. The results showed that there are no significant difference between prediction accuracies when different genetic operators used.

Parameters	Min accuracy	Max accuracy	Average accuracy	Average # of bireducts used (out of 400)	Average size of bireduct (o x a)
CX and RE	82.08	84.34	83.29	400	4018 x 18
OX and RE	82.08	84.34	83.04	400	4017 x 18
CX and Inversion	82.26	85.21	83.70	400	4017 x 18
OX and Inversion	83.65	85.73	84.77	400	4018 x 18

Table 4.8: Prediction Accuracy Comparison. Pareto ranking. Spam dataset

Table 4.8 shows the prediction accuracies, number of bireducts used and the sizes of bireducts for Pareto ranking fitness evaluation method and Spam dataset. OX and Inversion gave us the best prediction accuracy using Pareto ranking. One-way ANOVA test results with the significance level of 0.05 for Spam dataset are shown in TableB.6. It showed that OX and Inversion gave us the prediction accuracy that is significantly better compared to other genetic operators used.

4.3.4 Analysis

Parameters	Min accuracy	Max accuracy	Average accuracy	Average # of bireducts used (out of 300)	Average size of bireduct (o x a)
Weighted sum(7,1.5), CX and RE	78.72	100	93.90	101	475 x 9
Sum of ranks, CX and RE	95.74	98.93	97.58	300	474 x 12
Pareto ranking, CX and Inversion	95.74	98.93	97.62	300	474 x 12

Table 4.9: Prediction Accuracy Comparison. Pareto ranking. WDBC dataset

Table 4.9 shows the prediction accuracies, number of bireducts used and the sizes of bireducts for the best results of different fitness evaluation method and WDBC dataset. It was shown that Pareto ranking fitness evaluation gave the best prediction accuracy, while weighted sum gave the worst. One-way ANOVA test results with the significance level of 0.05 for Spam dataset are shown in Table B.7. It showed that weighted sum fitness evaluation approach gave the prediction accuracy significantly worse then two other strategies.

Parameters	Min accuracy	Max accuracy	Average accuracy	Average # of bireducts used (out of 400)	Average size of bireduct (o x a)
Weighted sum(1, 0.5), OX and Inversion	86.43	88.69	87.53	4	4024 x 14
Sum of ranks, OX and Inversion	82.78	85.04	84.04	400	4018 x 18
Pareto ranking, OX and Inversion	83.65	85.73	84.77	400	4018 x 18

Table 4.10: Prediction Accuracy Comparison. Pareto ranking. Spam dataset

Table 4.10 shows the prediction accuracies, number of bireducts used and the sizes of bireducts for the best results of different fitness evaluation method and Spam dataset. It was shown that weighted sum fitness evaluation strategy gave the best prediction accuracy, while Sum of ranks gave the worst. One-way ANOVA test results with the significance level of 0.05 for Spam dataset are shown in Table B.8. It showed that all the results are significantly different.

4.4 Accuracy comparison of proposed method with other known results.

Table 4.11 shows the average prediction accuracy of our proposed method and results of other known methods. From the table below it can be inherited that the results received by our MOGA system outperformed on SPECT, WDBC and Spam datasets the other methods found in machine learning literature. The proposed method gave the worst prediction accuracy for Pima datasets compared to other methods.

	Proposed method	Wang *et al.* [34]	GAP (avg.)	C4.5 (J48)	CLIP3	Canonical GP	ALPS	FSALPS
WDBC	**97.62**	95.6 [34]	96.14 [32]	93.88 [32]	-	93.12 [1]	92.80 [1]	93.82 [1]
Spam	**87.53**	86.9 [34]	-	-	-	-	-	-
SPECT	**99.80**	-	-	-	83.08 [16]	-	-	-
Ionosphere	86.55	87.80 [34]	89.90 [32]	89.82 [32]	-	88.02 [1]	89.71 [1]	**90.65** [1]
Pima	65.21	**77.3** [34]	-	-	-	73.47 [1]	73.81 [1]	74.03 [1]

Table 4.11: Prediction Accuracy Comparison.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Following the work started by Ślęzak and Janusz [30], we proposed a new approach to generate bireducts using a multi–objective genetic algorithm. Compared to the research mentioned before, there was no need to provide the ratio value to the system, change it and generate a huge amount of bireducts with different attributes/objects ratios. We were able to reduce the number of bireducts necessary in order to receive a good prediction accuracy, by using better quality bireducts provided by MOGA.

Our MOGA System estimated a quality of each bireduct by values of two objective functions as evolution progressed. As a result we received a set of bireducts with optimized values of these objectives. Different fitness evaluation methods and genetic operators, such as crossover and mutation were applied and the prediction accuracies were compared. Five datasets from UCI Machine Learning Repository [17] were used to test the proposed method and two datasets were used to perform a comparison study of different genetic operators and fitness evaluation strategies. Statistical analysis using one-way ANOVA test was performed to determine the significant difference between the results.

It was shown that in most cases different crossover and mutation operators do not significantly affect the prediction accuracy for tested datasets. Also it was shown that crossover and mutation operators affect prediction accuracy for different datasets in a different way.

The experiments showed that the fitness evaluation strategy affects the prediction accuracy in more significant way compared to the genetic operators. It was observed that the strategy that gave the best prediction accuracy depends on the dataset used.

The prediction accuracies of the proposed method are comparable with the best

results in machine learning literature [1, 16, 34]. The results received by our MOGA system outperformed on SPECT, WDBC and Spam datasets the methods mentioned in [1, 16, 34].

5.2 Future Work

Although the initial results are encouraging, further analysis should be done using various data sets larger in size.

The main focus of our work was to carry out an empirical study comparing different fitness evaluation techniques, as well as incorporating further genetic operators like mutation and crossover. We did not pay enough attention to the study of different discretization methods, as well as different ways of dealing with missing values which in theory can significantly skew the results.

In our current research, our RS system generates the decision rules independently based on each bireduct and applies to the entire testing dataset. Eventually, the number of support cases are calculated for each outcome. Another way of dealing with the rules that should be tested is the use of decision trees. The idea is to generate a decision tree based on all bireducts and then apply to the testing dataset at once.

Bibliography

[1] Awuley, A. (2015) Feature Selection and Classification Using Age Layered Population Structure Genetic Programming. Master's thesis, Brock University.

[2] Bazan, J. G., Nguyen, H. S., Nguyen, S. H., Synak, P., and Wróblewski, J. (2000) Rough set methods and applications. chap. Rough Set Algorithms in Classification Problem, pp. 49–88, Physica-Verlag GmbH, Heidelberg, Germany.

[3] Bentley, P. J. and Wakefield, J. P. (1998) Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. Chawdhry, P. K., Roy, R., and Pant, R. K. (eds.), *Soft Computing in Engineering Design and Manufacturing*, pp. 231–240, Springer-Verlag.

[4] Bjorvand, A. T. and Komorowski, J. (1997) Practical applications of genetic algorithms for efficient reduct computation. *In Proc 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, **4**, 601–606.

[5] Branke, J., Deb, K., Miettinen, K., and Slowinski, R. (eds.) (2008) *Multiobjective Optimization, Interactive and Evolutionary Approaches*, vol. 5252 of *Lecture Notes in Computer Science*, Springer.

[6] Darwin, C. (1859) *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. John Murray, London.

[7] Davis, L. (1985) Applying adaptive algorithms to epistatic domains. *Proceedings of the 9th International Joint Conference on Artificial Intelligence - Volume 1*, pp. 162–164, IJCAI'85, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[8] Deb, K. (2001) *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, Chichester, UK.

[9] Düntsch, I. and Gediga, G. (2000) *Rough set data analysis: A road to non-invasive knowledge discovery*. Methoδos Primers Series, Bangor, UK.

[10] Frank, E. and Kramer, S. (2004) Ensembles of nested dichotomies for multi-class problems. *In Proc 21st International Conference on Machine Learning*, pp. 305–312, ACM Press.

[11] Gennari, J. H., Langley, P., and Fisher, D. (1989) Models of incremental concept formation. *Artif. Intell.*, **40**, 11–61.

[12] Georg Peters, R. T. (2012) *Workflow Management Supported by Rough Set Concepts*. Springer-Verlag London Limited.

[13] Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

[14] Holland, J. H. (1992) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.

[15] Jan Komorowski, L. P., Zdzislaw Pawlak and Skowron, A. (1998), Rough sets: A tutorial. Http://www.lcb.uu.se/users/janko/data/rstutorial.pdf.

[16] Kurgan, L. A., Cios, K. J., Tadeusiewicz, R., Ogiela, M., and Goodenday, L. S. (2001) Knowledge discovery approach to automated cardiac SPECT diagnosis. *Artificial Intelligence in Medicine*, **23**, 149 – 169.

[17] Lichman, M. (2013), UCI machine learning repository. Http://archive.ics.uci.edu/ml.

[18] McKee, T. E. and Lensberg, T. (2002) Genetic programming and rough sets: A hybrid approach to bankruptcy classification. *European Journal of Operational Research*, **138**, 436–451.

[19] Mendel, G. (1865) Experiments in Plant Hybridization. *Proceedings of the Brünn Natural History Society*, **IV**.

[20] Moshkov, M. J., Piliszczuk, M., and Zielosko, B. (2008) *Partial Covers, Reducts and Decision Rules in Rough Sets - Theory and Applications*, vol. 145 of *Studies in Computational Intelligence*. Springer.

[21] Nowak-Brzezinska, A., Rough set theory in decision support systems. Http://zsi.tech.us.edu.pl/∼nowak/bien/w2.pdf.

[22] Oliver, I. M., Smith, D. J., and Holland, J. R. C. (1987) A study of permutation crossover operators on the traveling salesman problem. *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pp. 224–230, L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

[23] Ombuki, B., J. Ross, B., and Hanshar, F. (2006) Multi-objective genetic algorithm for vehicle routing problem with time windows. *Applied Intelligence*, **24**, 17–30.

[24] Pawlak, Z. (1982) Rough sets. *International Journal of Information and Computer Sciences*, **11**, 341–356.

[25] Polkowski, L. (2002) *Rough Sets: Mathematical Foundations*. Advances in Intelligent and Soft Computing, Physica-Verlag HD.

[26] Sikora, M. and Sikora, B. (2012) *Rough Natural Hazards Monitoring*. Springer-Verlag London.

[27] Skowron, A. and Rauszer, C. (1992) The discernibility matrices and functions in information systems. Słowiński, R. (ed.), *Intelligent decision support: Handbook of applications and advances of rough set theory*, vol. 11 of *System Theory, Knowledge Engineering and Problem Solving*, pp. 331–362, Kluwer, Dordrecht.

[28] Ślęzak, D. (2009) Rough sets and functional dependencies in data: Foundations of association reducts. Gavrilova, M., Tan, C., Wang, Y., and Chan, K. (eds.), *Transactions on Computational Science V*, vol. 5540 of *Lecture Notes in Computer Science*, pp. 182–205, Springer Verlag.

[29] Ślęzak, D. (2009) Rough sets and functional dependencies in data: Foundations of association reducts. *Transactions on Computational Science*, **5**, 182–205.

[30] Ślęzak, D. and Janusz, A. (2011) Ensembles of bireducts: Towards robust classification and simple representation. *Proc. of FGIT 2011*, **7105**, 64–77.

[31] Slezak, D. and Widz, S. (2010) Is it important which rough-set-based classifier extraction and voting criteria are applied together? Szczuka, M. S., Kryszkiewicz,

M., Ramanna, S., Jensen, R., and Hu, Q. (eds.), *RSCTC*, vol. 6086 of *Lecture Notes in Computer Science*, pp. 187–196, Springer.

[32] Smith, M. and Bull, L. (2004) Using genetic programming for feature creation with a genetic algorithm feature selector. Yao, X., Burke, E., Lozano, J., Smith, J., Merelo-Guervs, J., Bullinaria, J., Rowe, J., Tio, P., Kabn, A., and Schwefel, H.-P. (eds.), *Parallel Problem Solving from Nature - PPSN VIII*, vol. 3242 of *Lecture Notes in Computer Science*, pp. 1163–1171, Springer Berlin Heidelberg.

[33] Stawicki, S. and Slezak, D. (2013) Recent advances in decision bireducts: Complexity, heuristics and streams. Lingras, P., Wolski, M., Cornelis, C., Mitra, S., and Wasilewski, P. (eds.), *RSKT*, vol. 8171 of *Lecture Notes in Computer Science*, pp. 200–212, Springer.

[34] Wang, Y. and Witten, I. H. (2002) Modeling for optimal probability prediction. Sammut, C. and Hoffmann, A. G. (eds.), *ICML*, pp. 650–657, Morgan Kaufmann.

[35] Wroblewski, J. (1998) Genetic algorithms in decomposition and classification problems. Polkowski, L. and Skowron, A. (eds.), *Rough Sets in Knowledge Discovery 2*, vol. 19 of *Studies in Fuzziness and Soft Computing*, pp. 471–487, Physica-Verlag HD.

Appendix A

Number of objects and attributes in bireducts

the figures below illustrates the number of attributes and number of objects in each bireduct for each run done by using our MOGA system. The black border indicates the total number of objects and attributes in the training dataset. Each coloured dot indicates the size of bireduct (objects x attributes).

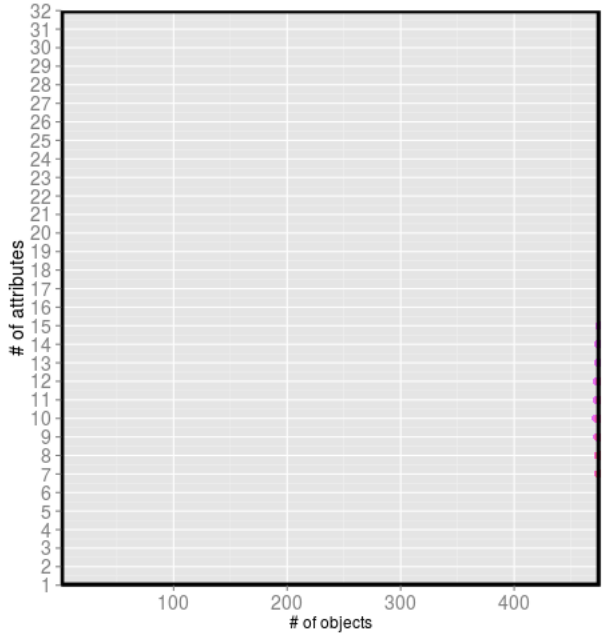

Figure A.1: Number of objects and attributes in bireducts for each run. Weighted Sum (0.7, 1.5), OX and RE. WDBC dataset

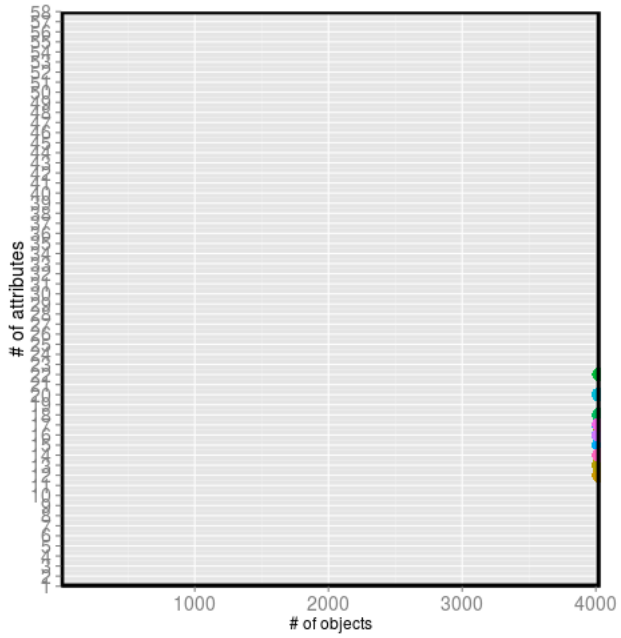

Figure A.2: Number of objects and attributes in bireducts for each run. Weighted Sum (0.7, 1.5), OX and RE. Spam dataset

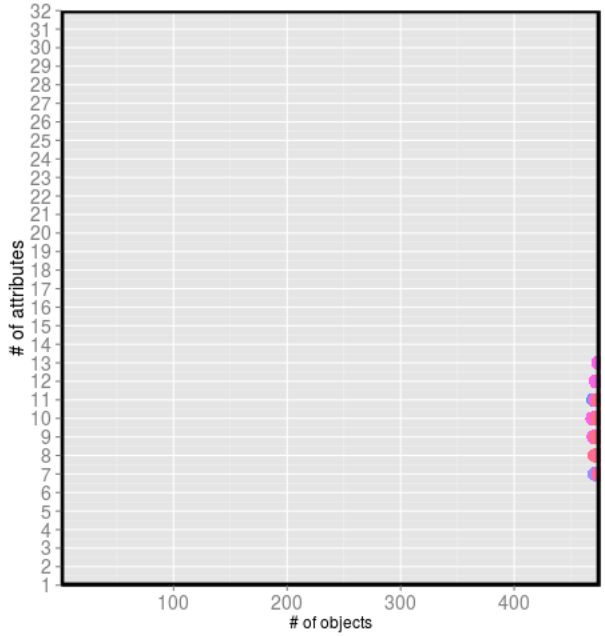

Figure A.3: Number of objects and attributes in bireducts for each run. Weighted Sum (1, 0.5), OX and RE. WDBC dataset

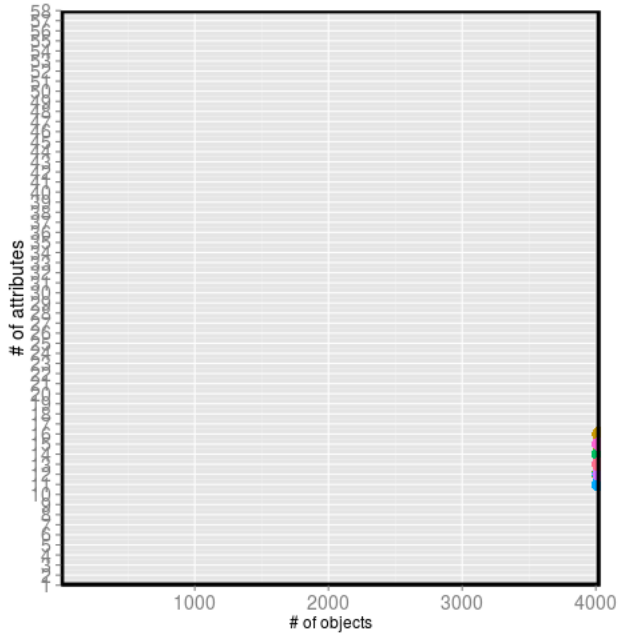

Figure A.4: Number of objects and attributes in bireducts for each run. Weighted Sum (1, 0.5), OX and RE. Spam dataset

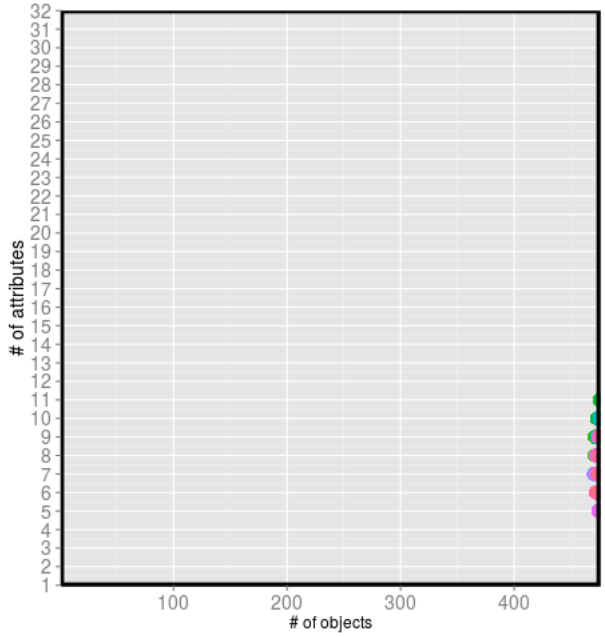

Figure A.5: Number of objects and attributes in bireducts for each run. Weighted Sum (1, 0.5), CX and RE. WDBC dataset

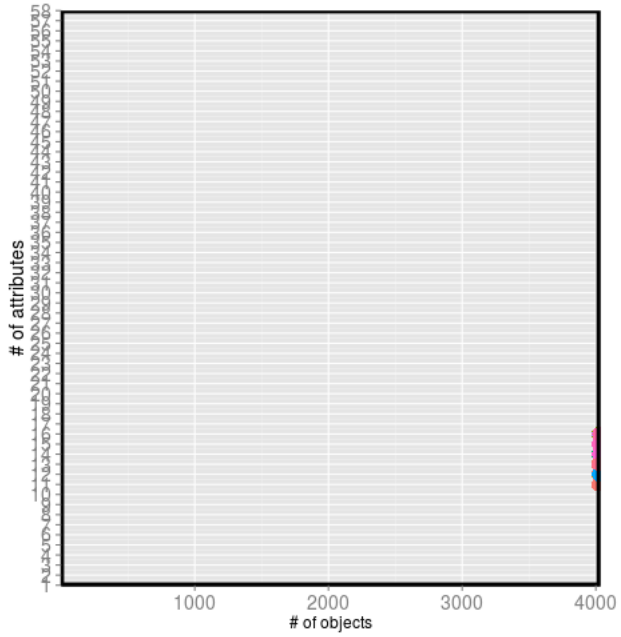

Figure A.6: Number of objects and attributes in bireducts for each run. Weighted Sum (1, 0.5), CX and RE. Spam dataset

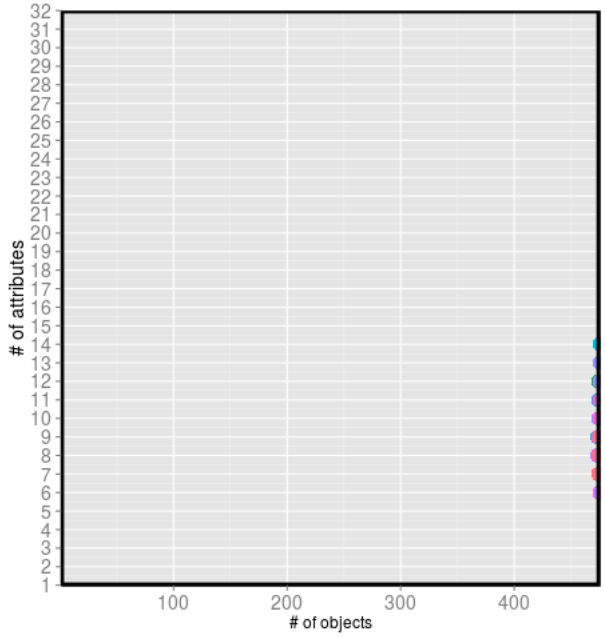

Figure A.7: Number of objects and attributes in bireducts for each run. Weighted Sum (0.7, 1.5), CX and RE. WDBC dataset

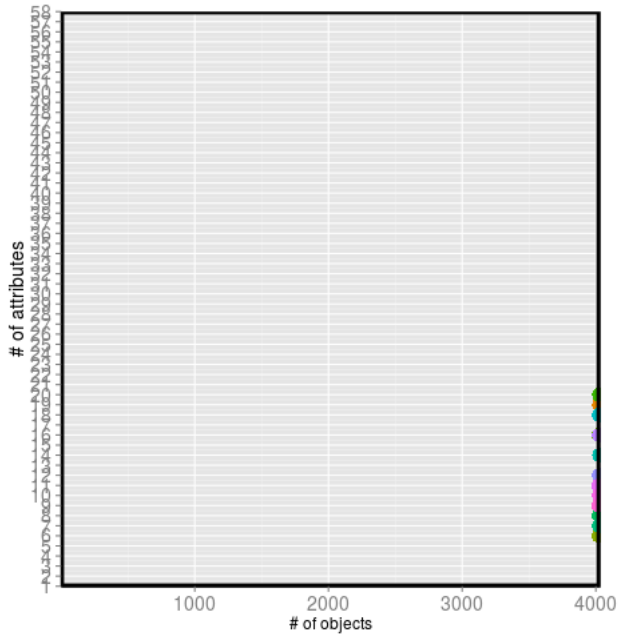

Figure A.8: Number of objects and attributes in bireducts for each run. Weighted Sum (0.7, 1.5), CX and RE. Spam dataset

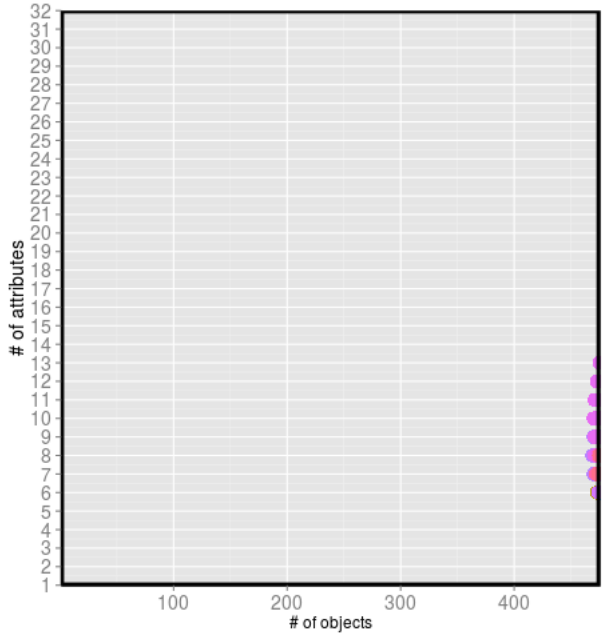

Figure A.9: Number of objects and attributes in bireducts for each run. Weighted Sum (1, 0.5), OX and Inversion. WDBC dataset

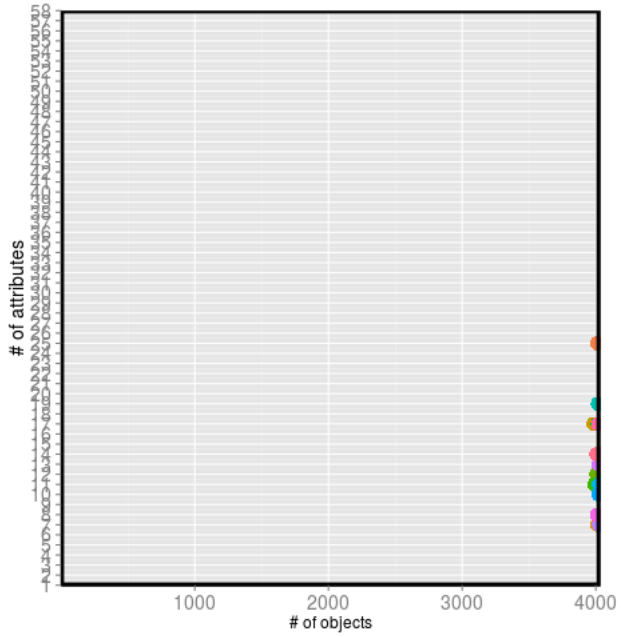

Figure A.10: Number of objects and attributes in bireducts for each run. Weighted Sum (1, 0.5), OX and Inversion. Spam dataset

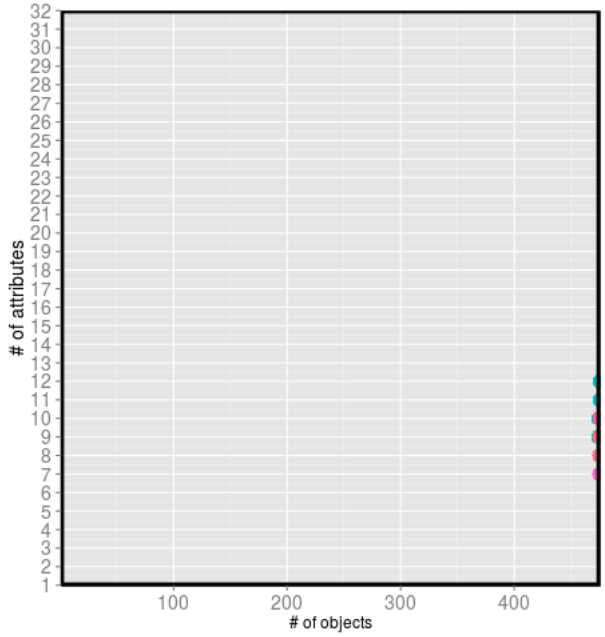

Figure A.11: Number of objects and attributes in bireducts for each run. Weighted Sum (0.7, 1.5), OX and Inversion. WDBC dataset

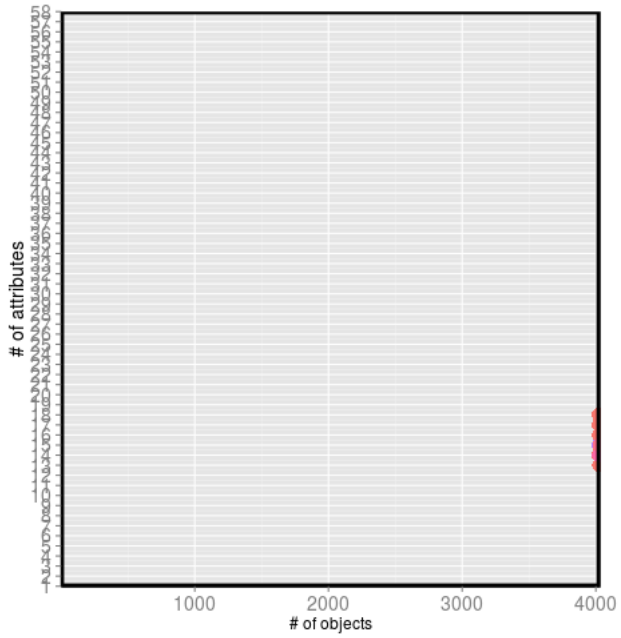

Figure A.12: Number of objects and attributes in bireducts for each run. Weighted Sum (0.7, 1.5), OX and Inversion. Spam dataset

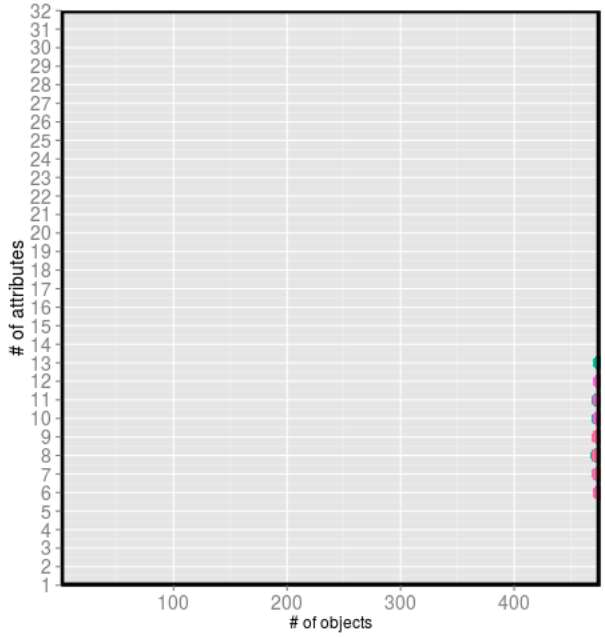

Figure A.13: Number of objects and attributes in bireducts for each run. Weighted Sum (0.7, 1.5), CX and Inversion. WDBC dataset

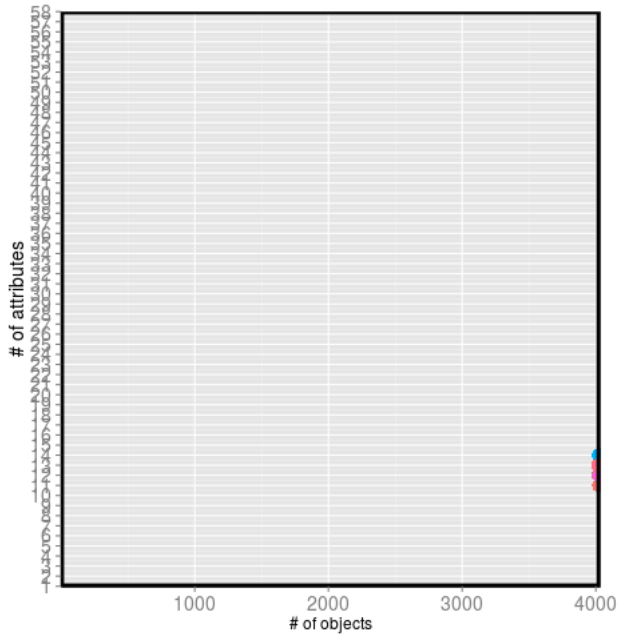

Figure A.14: Number of objects and attributes in bireducts for each run. Weighted Sum (0.7, 1.5), CX and Inversion. Spam dataset

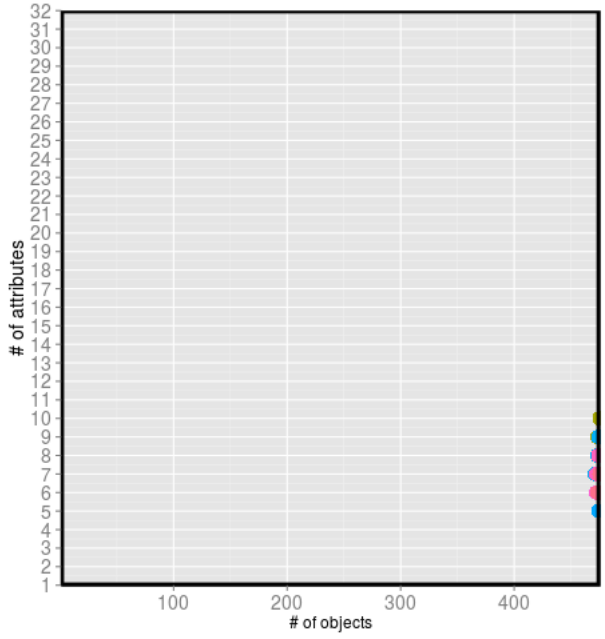

Figure A.15: Number of objects and attributes in bireducts for each run. Weighted Sum (1, 0.5), CX and Inversion. WDBC dataset

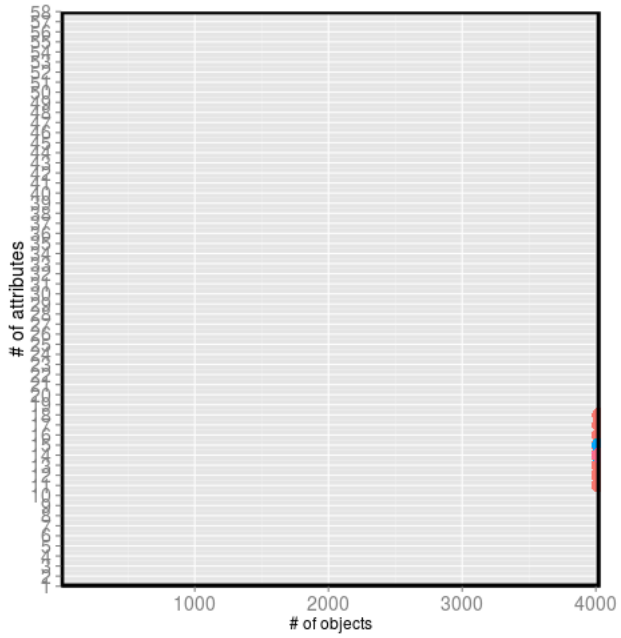

Figure A.16: Number of objects and attributes in bireducts for each run. Weighted Sum (1, 0.5), CX and Inversion. Spam dataset (1, 0.5)

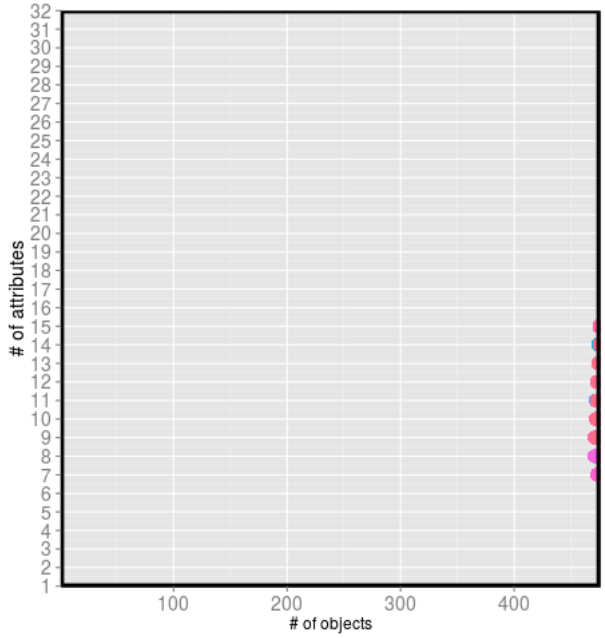

Figure A.17: Number of objects and attributes in bireducts for each run. Normalized Sum Of Ranks, OX and RE. WDBC dataset

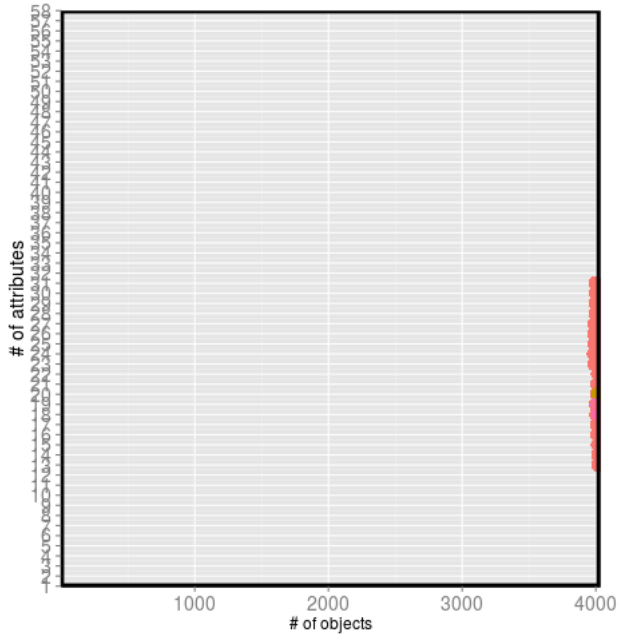

Figure A.18: Number of objects and attributes in bireducts for each run. Normalized Sum Of Ranks, OX and RE. Spam dataset

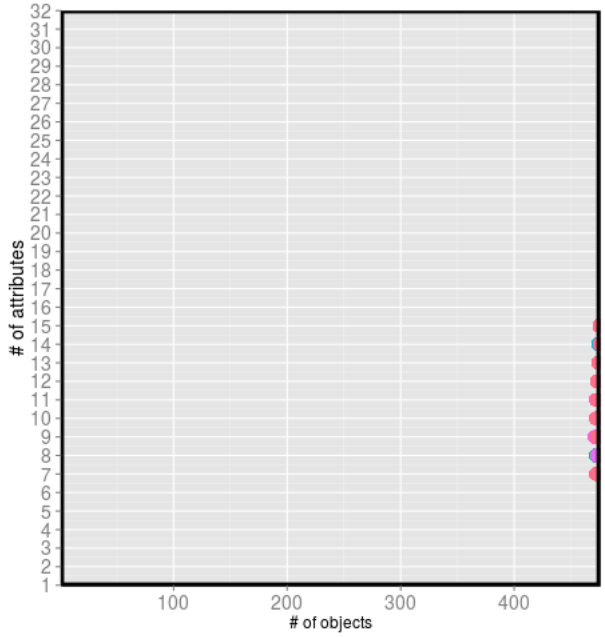

Figure A.19: Number of objects and attributes in bireducts for each run. Normalized Sum Of Ranks, CX and RE. WDBC dataset

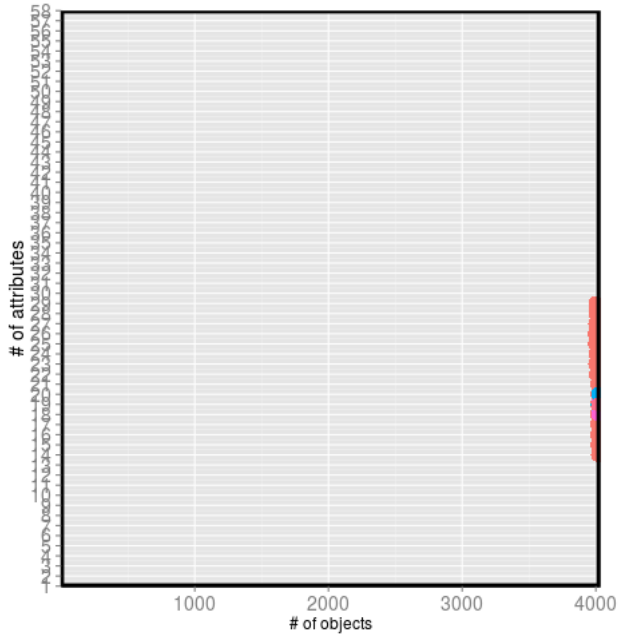

Figure A.20: Number of objects and attributes in bireducts for each run. Normalized Sum Of Ranks, CX and RE. Spam dataset

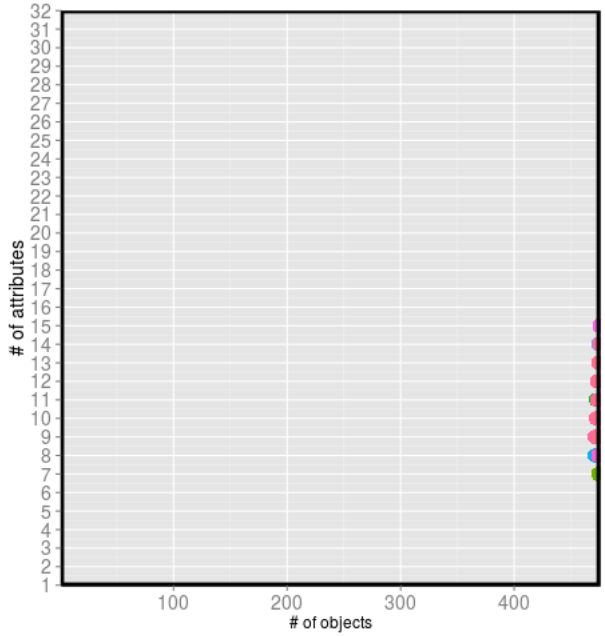

Figure A.21: Number of objects and attributes in bireducts for each run. Normalized Sum Of Ranks, OX and Inversion. WDBC dataset

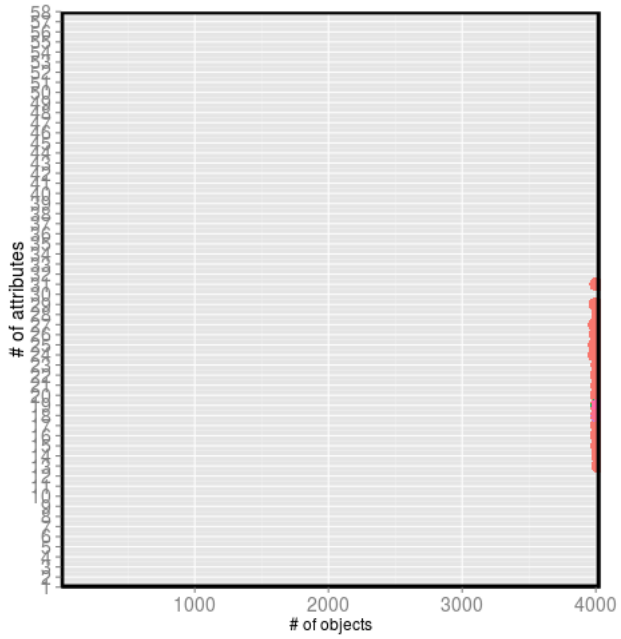

Figure A.22: Number of objects and attributes in bireducts for each run. Normalized Sum Of Ranks, OX and Inversion. Spam dataset

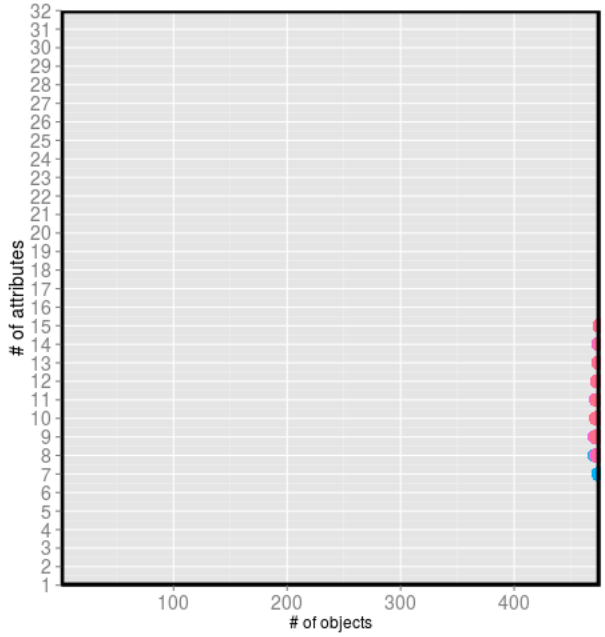

Figure A.23: Number of objects and attributes in bireducts for each run. Normalized Sum Of Ranks, CX and Inversion. WDBC dataset

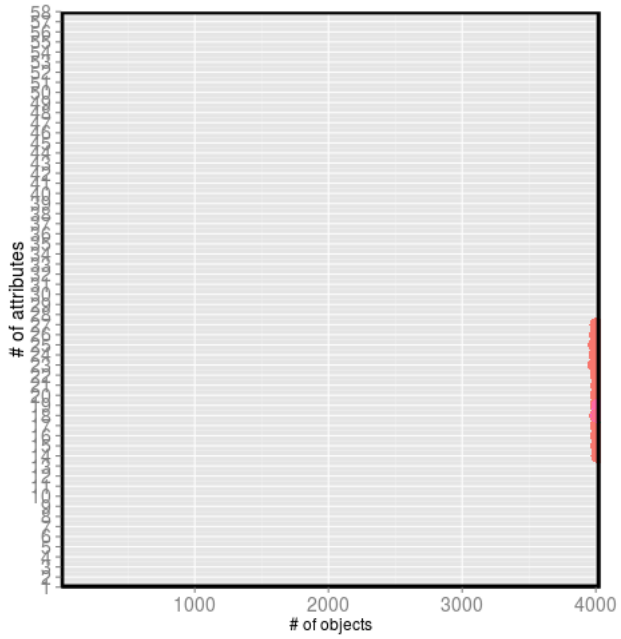

Figure A.24: Number of objects and attributes in bireducts for each run. Normalized Sum Of Ranks, CX and Inversion. Spam dataset

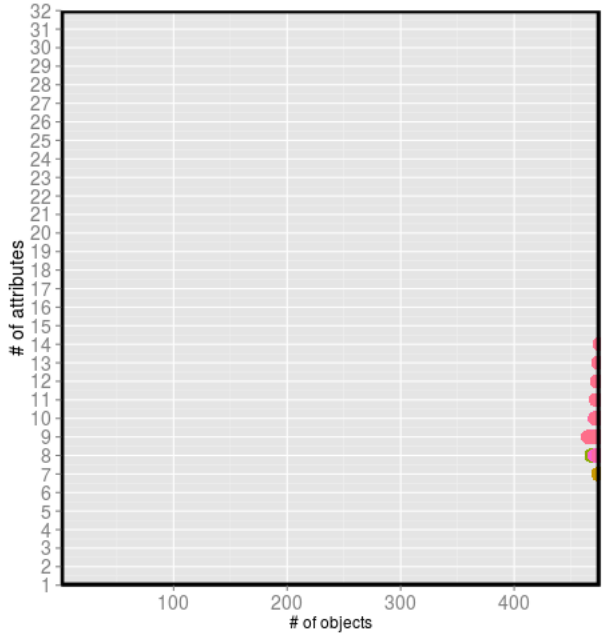

Figure A.25: Number of objects and attributes in bireducts for each run. Pareto ranking, OX and RE. WDBC dataset

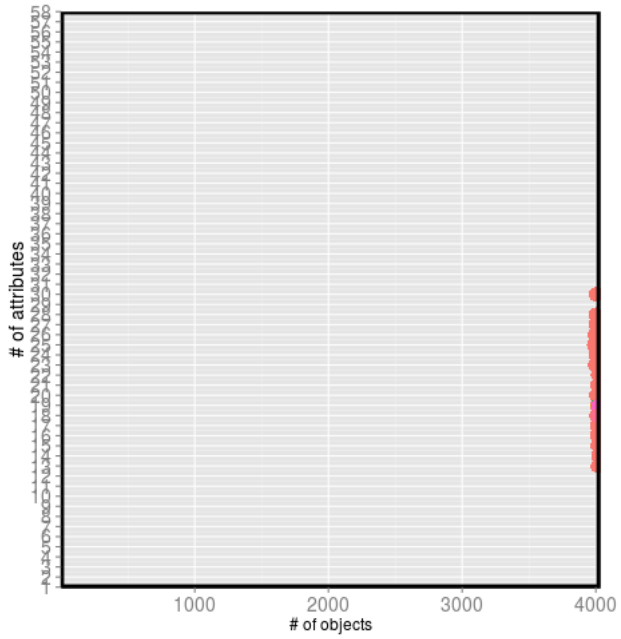

Figure A.26: Number of objects and attributes in bireducts for each run. Pareto ranking, OX and RE. Spam dataset

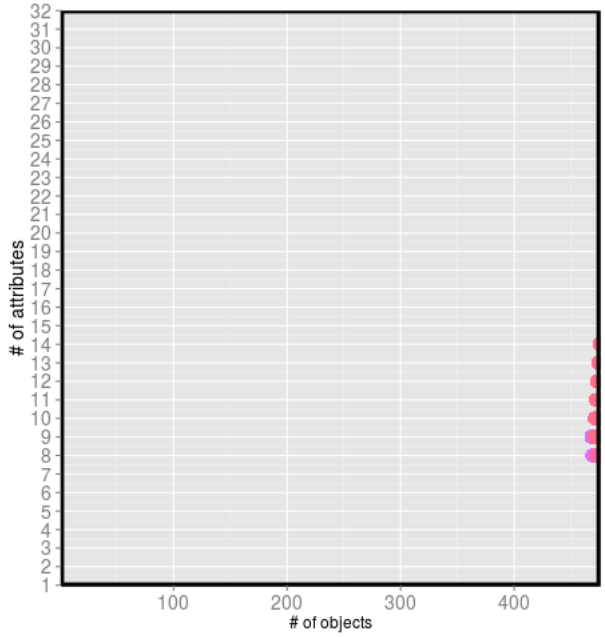

Figure A.27: Number of objects and attributes in bireducts for each run. Pareto ranking, CX and RE. WDBC dataset

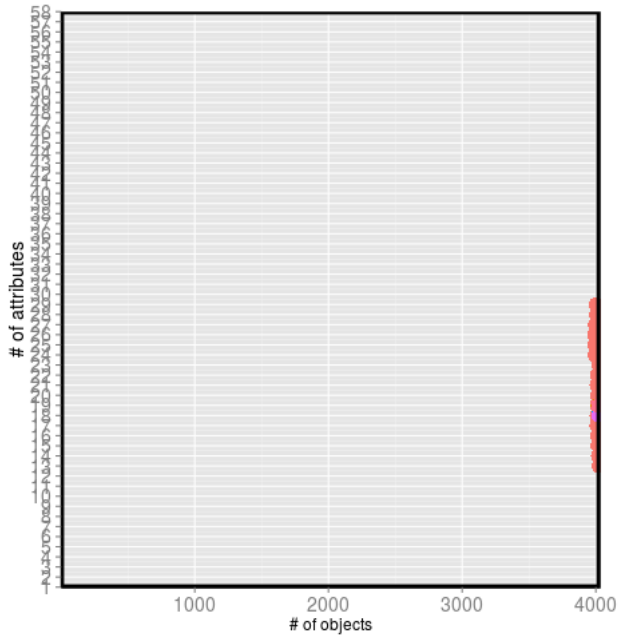

Figure A.28: Number of objects and attributes in bireducts for each run. Pareto ranking, CX and RE. Spam dataset

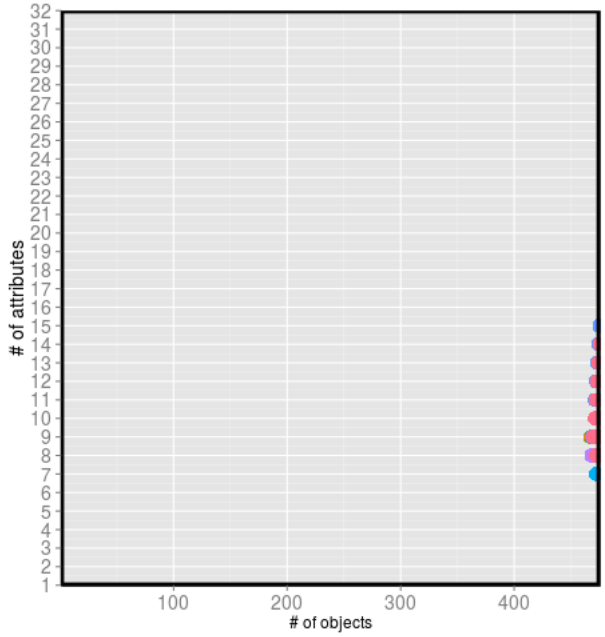

Figure A.29: Number of objects and attributes in bireducts for each run. Pareto ranking, OX and Inversion. WDBC dataset

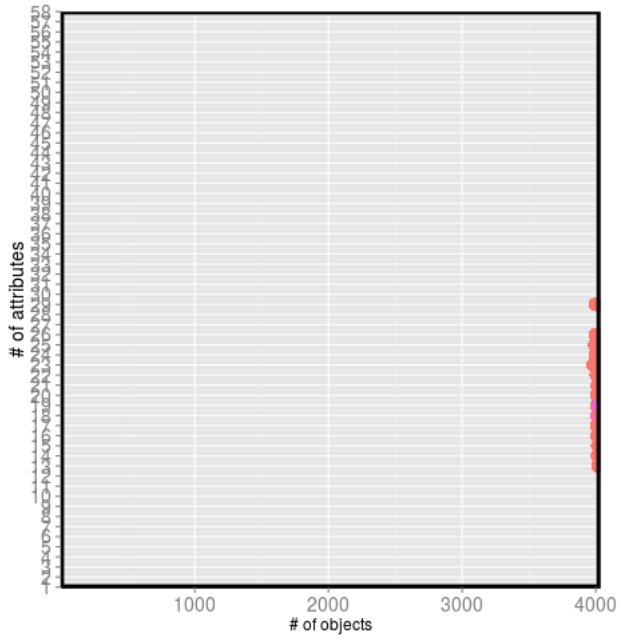

Figure A.30: Number of objects and attributes in bireducts for each run. Pareto ranking, OX and Inversion. Spam dataset

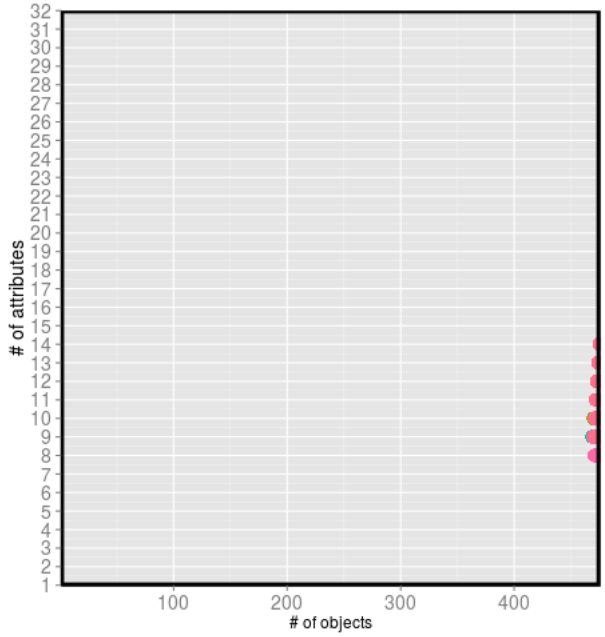

Figure A.31: Number of objects and attributes in bireducts for each run. Pareto ranking, CX and Inversion. WDBC dataset

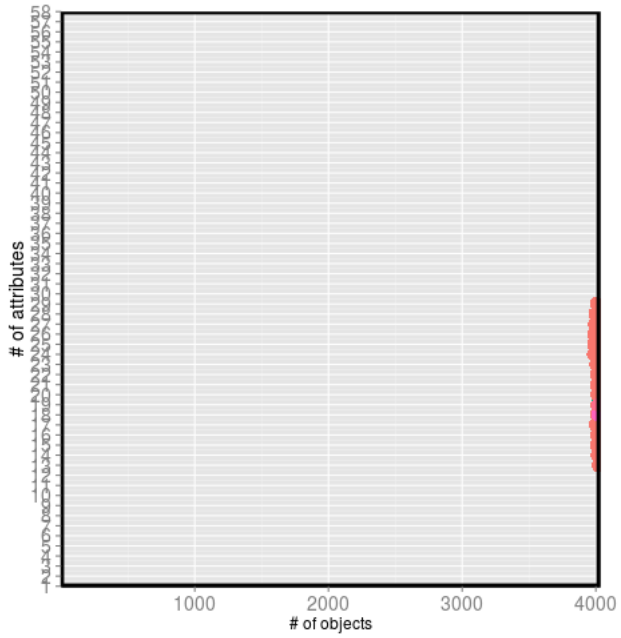

Figure A.32: Number of objects and attributes in bireducts for each run. Pareto ranking, CX and Inversion. Spam dataset

Appendix B

One-way ANOVA test

In this test the results of one-way ANOVA test with the significance level of 0.05 are shown. The significance values that are less then 0.05 are coloured in blue and show that the difference between corresponding methods is significant.

(I) factor	(J) factor	Sig.
Weighted Sum(0.7,1.5), CX and RE	Weighted Sum(1,0.5), CX and RE	.057
	Weighted Sum(0.7,1.5), OX and RE	1.000
	Weighted Sum(1,0.5), OX and RE	.886
	Weighted Sum(0.7,1.5), CX and Inversion	.601
	Weighted Sum(1,0.5), CX and Inversion	.316
	Weighted Sum(0.7,1.5), OX and Inversion	.999
	Weighted Sum(1,0.5), OX and Inversion	.414
Weighted Sum(1,0.5), CX and RE	Weighted Sum(0.7,1.5), CX and RE	.057
	Weighted Sum(0.7,1.5), OX and RE	.064
	Weighted Sum(1,0.5), OX and RE	.699
	Weighted Sum(0.7,1.5), CX and Inversion	.936
	Weighted Sum(1,0.5), CX and Inversion	.995
	Weighted Sum(0.7,1.5), OX and Inversion	.219
	Weighted Sum(1,0.5), OX and Inversion	.985
Weighted Sum(0.7,1.5), OX and RE	Weighted Sum(0.7,1.5), CX and RE	1.000
	Weighted Sum(1,0.5), CX and RE	.064
	Weighted Sum(1,0.5), OX and RE	.903
	Weighted Sum(0.7,1.5), CX and Inversion	.631
	Weighted Sum(1,0.5), CX and Inversion	.341
	Weighted Sum(0.7,1.5), OX and Inversion	1.000
	Weighted Sum(1,0.5), OX and Inversion	.442
Weighted Sum(1,0.5), OX and RE	Weighted Sum(0.7,1.5), CX and RE	.886
	Weighted Sum(1,0.5), CX and RE	.699
	Weighted Sum(0.7,1.5), OX and RE	.903
	Weighted Sum(0.7,1.5), CX and Inversion	1.000
	Weighted Sum(1,0.5), CX and Inversion	.982
	Weighted Sum(0.7,1.5), OX and Inversion	.994
	Weighted Sum(1,0.5), OX and Inversion	.994
Weighted Sum(0.7,1.5), CX and Inversion	Weighted Sum(0.7,1.5), CX and RE	.601
	Weighted Sum(1,0.5), CX and RE	.936
	Weighted Sum(0.7,1.5), OX and RE	.631
	Weighted Sum(1,0.5), OX and RE	1.000
	Weighted Sum(1,0.5), CX and Inversion	1.000
	Weighted Sum(0.7,1.5), OX and Inversion	.908
	Weighted Sum(1,0.5), OX and Inversion	1.000
Weighted Sum(1,0.5), CX and Inversion	Weighted Sum(0.7,1.5), CX and RE	.316
	Weighted Sum(1,0.5), CX and RE	.995
	Weighted Sum(0.7,1.5), OX and RE	.341
	Weighted Sum(1,0.5), OX and RE	.982
	Weighted Sum(0.7,1.5), CX and Inversion	1.000
	Weighted Sum(0.7,1.5), OX and Inversion	.680
	Weighted Sum(1,0.5), OX and Inversion	1.000
Weighted Sum(0.7,1.5), OX and Inversion	Weighted Sum(0.7,1.5), CX and RE	.999
	Weighted Sum(1,0.5), CX and RE	.219
	Weighted Sum(0.7,1.5), OX and RE	1.000
	Weighted Sum(1,0.5), OX and RE	.994
	Weighted Sum(0.7,1.5), CX and Inversion	.908
	Weighted Sum(1,0.5), CX and Inversion	.680
	Weighted Sum(1,0.5), OX and Inversion	.779
Weighted Sum(1,0.5), OX and Inversion	Weighted Sum(0.7,1.5), CX and RE	.414
	Weighted Sum(1,0.5), CX and RE	.985
	Weighted Sum(0.7,1.5), OX and RE	.442
	Weighted Sum(1,0.5), OX and RE	.994
	Weighted Sum(0.7,1.5), CX and Inversion	1.000
	Weighted Sum(1,0.5), CX and Inversion	1.000
	Weighted Sum(0.7,1.5), OX and Inversion	.779

Table B.1: One-way ANOVA test. Weighted Sum. WDBC dataset.

(I) factor	(J) factor	Sig.
Weighted Sum(0.7,1.5), CX and RE	Weighted Sum(1,0.5), CX and RE	.003
	Weighted Sum(0.7,1.5), OX and RE	.000
	Weighted Sum(1,0.5), OX and RE	.000
	Weighted Sum(0.7,1.5), CX and Inversion	.218
	Weighted Sum(1,0.5), CX and Inversion	.001
	Weighted Sum(0.7,1.5), OX and Inversion	.000
	Weighted Sum(1,0.5), OX and Inversion	.000
Weighted Sum(1,0.5), CX and RE	Weighted Sum(0.7,1.5), CX and RE	.003
	Weighted Sum(0.7,1.5), OX and RE	.152
	Weighted Sum(1,0.5), OX and RE	.945
	Weighted Sum(0.7,1.5), CX and Inversion	.000
	Weighted Sum(1,0.5), CX and Inversion	.000
	Weighted Sum(0.7,1.5), OX and Inversion	.000
	Weighted Sum(1,0.5), OX and Inversion	.000
Weighted Sum(0.7,1.5), OX and RE	Weighted Sum(0.7,1.5), CX and RE	.000
	Weighted Sum(1,0.5), CX and RE	.152
	Weighted Sum(1,0.5), OX and RE	.819
	Weighted Sum(0.7,1.5), CX and Inversion	.000
	Weighted Sum(1,0.5), CX and Inversion	.000
	Weighted Sum(0.7,1.5), OX and Inversion	.000
	Weighted Sum(1,0.5), OX and Inversion	.000
Weighted Sum(1,0.5), OX and RE	Weighted Sum(0.7,1.5), CX and RE	.000
	Weighted Sum(1,0.5), CX and RE	.945
	Weighted Sum(0.7,1.5), OX and RE	.819
	Weighted Sum(0.7,1.5), CX and Inversion	.000
	Weighted Sum(1,0.5), CX and Inversion	.000
	Weighted Sum(0.7,1.5), OX and Inversion	.000
	Weighted Sum(1,0.5), OX and Inversion	.000
Weighted Sum(0.7,1.5), CX and Inversion	Weighted Sum(0.7,1.5), CX and RE	.218
	Weighted Sum(1,0.5), CX and RE	.000
	Weighted Sum(0.7,1.5), OX and RE	.000
	Weighted Sum(1,0.5), OX and RE	.000
	Weighted Sum(1,0.5), CX and Inversion	.725
	Weighted Sum(0.7,1.5), OX and Inversion	.000
	Weighted Sum(1,0.5), OX and Inversion	.000
Weighted Sum(1,0.5), CX and Inversion	Weighted Sum(0.7,1.5), CX and RE	.001
	Weighted Sum(1,0.5), CX and RE	.000
	Weighted Sum(0.7,1.5), OX and RE	.000
	Weighted Sum(1,0.5), OX and RE	.000
	Weighted Sum(0.7,1.5), CX and Inversion	.725
	Weighted Sum(0.7,1.5), OX and Inversion	.000
	Weighted Sum(1,0.5), OX and Inversion	.000
Weighted Sum(0.7,1.5), OX and Inversion	Weighted Sum(0.7,1.5), CX and RE	.000
	Weighted Sum(1,0.5), CX and RE	.000
	Weighted Sum(0.7,1.5), OX and RE	.000
	Weighted Sum(1,0.5), OX and RE	.000
	Weighted Sum(0.7,1.5), CX and Inversion	.000
	Weighted Sum(1,0.5), CX and Inversion	.000
	Weighted Sum(1,0.5), OX and Inversion	.945
Weighted Sum(1,0.5), OX and Inversion	Weighted Sum(0.7,1.5), CX and RE	.000
	Weighted Sum(1,0.5), CX and RE	.000
	Weighted Sum(0.7,1.5), OX and RE	.000
	Weighted Sum(1,0.5), OX and RE	.000
	Weighted Sum(0.7,1.5), CX and Inversion	.000
	Weighted Sum(1,0.5), CX and Inversion	.000
	Weighted Sum(0.7,1.5), OX and Inversion	.945

Table B.2: One-way ANOVA test. Weighted Sum. Spam dataset.

(I) factor	(J) factor	Sig.
Normalized Sum Of Ranks, CX and RE	Normalized Sum Of Ranks, OX and RE	.951
	Normalized Sum Of Ranks, CX and Inversion	.855
	Normalized Sum Of Ranks, OX and Inversion	.993
Normalized Sum Of Ranks, OX and RE	Normalized Sum Of Ranks, CX and RE	.951
	Normalized Sum Of Ranks, CX and Inversion	.993
	Normalized Sum Of Ranks, OX and Inversion	.993
Normalized Sum Of Ranks, CX and Inversion	Normalized Sum Of Ranks, CX and RE	.855
	Normalized Sum Of Ranks, OX and RE	.993
	Normalized Sum Of Ranks, OX and Inversion	.951
Normalized Sum Of Ranks, OX and Inversion	Normalized Sum Of Ranks, CX and RE	.993
	Normalized Sum Of Ranks, OX and RE	.993
	Normalized Sum Of Ranks, CX and Inversion	.951

Table B.3: One-way ANOVA test. Normalized Sum Of Ranks. WDBC dataset.

(I) factor	(J) factor	Sig.
Normalized Sum Of Ranks, CX and RE	Normalized Sum Of Ranks, OX and RE	.207
	Normalized Sum Of Ranks, CX and Inversion	.622
	Normalized Sum Of Ranks, OX and Inversion	.716
Normalized Sum Of Ranks, OX and RE	Normalized Sum Of Ranks, CX and RE	.207
	Normalized Sum Of Ranks, CX and Inversion	.874
	Normalized Sum Of Ranks, OX and Inversion	.016
Normalized Sum Of Ranks, CX and Inversion	Normalized Sum Of Ranks, CX and RE	.622
	Normalized Sum Of Ranks, OX and RE	.874
	Normalized Sum Of Ranks, OX and Inversion	.112
Normalized Sum Of Ranks, OX and Inversion	Normalized Sum Of Ranks, CX and RE	.716
	Normalized Sum Of Ranks, OX and RE	.016
	Normalized Sum Of Ranks, CX and Inversion	.112

Table B.4: One-way ANOVA test. Normalized Sum Of Ranks. Spam dataset.

(I) factor_pr	(J) factor_pr	Sig.
Pareto ranking, CX and RE	Pareto ranking, OX and RE	.932
	Pareto ranking, CX and Inversion	.999
	Pareto ranking, OX and Inversion	.999
Pareto ranking, OX and RE	Pareto ranking, CX and RE	.932
	Pareto ranking, CX and Inversion	.876
	Pareto ranking, OX and Inversion	.876
Pareto ranking, CX and Inversion	Pareto ranking, CX and RE	.999
	Pareto ranking, OX and RE	.876
	Pareto ranking, OX and Inversion	1.000
Pareto ranking, OX and Inversion	Pareto ranking, CX and RE	.999
	Pareto ranking, OX and RE	.876
	Pareto ranking, CX and Inversion	1.000

Table B.5: One-way ANOVA test. Pareto ranking. WDBC dataset.

(I) factor	(J) factor	Sig.
Pareto ranking, CX and RE	Pareto ranking, OX and RE	.857
	Pareto ranking, CX and Inversion	.059
	Pareto ranking, OX and Inversion	.000
Pareto ranking, OX and RE	Pareto ranking, CX and RE	.857
	Pareto ranking, CX and Inversion	.006
	Pareto ranking, OX and Inversion	.000
Pareto ranking, CX and Inversion	Pareto ranking, CX and RE	.059
	Pareto ranking, OX and RE	.006
	Pareto ranking, OX and Inversion	.000
Pareto ranking, OX and Inversion	Pareto ranking, CX and RE	.000
	Pareto ranking, OX and RE	.000
	Pareto ranking, CX and Inversion	.000

Table B.6: One-way ANOVA test. Pareto ranking. Spam dataset.

(I) factor	(J) factor	Sig.
Weighted sum(0.7,1.5), CX and RE	Sum of ranks, CX and RE	.000
	Pareto ranking, CX and Inversion	.000
Sum of ranks, CX and RE	Weighted sum(0.7,1.5), CX and RE	.000
	Pareto ranking, CX and Inversion	.999
Pareto ranking, CX and Inversion	Weighted sum(0.7,1.5), CX and RE	.000
	Sum of ranks, CX and RE	.999

Table B.7: One-way ANOVA test for WDBC dataset. Best results for each fitness evaluation strategy.

(I) factor	(J) factor	Sig.
Weighted sum(1,0.5), OX and Inversion	Sum of ranks, OX and Inversion	.000
	Pareto ranking, OX and Inversion	.000
Sum of ranks, OX and Inversion	Weighted sum(1,0.5), OX and Inversion	.000
	Pareto ranking, OX and Inversion	.023
Pareto ranking, OX and Inversion	Weighted sum(1,0.5), OX and Inversion	.000
	Sum of ranks, OX and Inversion	.023

Table B.8: One-way ANOVA test for Spam dataset. Best results for each fitness evaluation strategy.

Appendix C

MOGA System Output Examples

This Appendix displays several examples of the proposed MOGA System bireduct output for each dataset.

Spambase dataset:

attributes:

2 4 5 9 11 18 20 26 27 49 51 54 55 56

objects:

4026 4025 4024 4023 4022 4021 4020 4019 4018 4017 4016 4015 4014 4013 4012 4011 4010 4009 4008 4007 4006 4005 4004 4003 4002 4001 4000 3999 3998 3997 3996 3995 3994 3993 3992 3991 3990 3989 3988 3987 3986 3985 3984 3983 3982 3981 3980 3979 3978 3977 3976 3975 3974 3973 3972 3971 3970 3969 3968 3967 3966 3965 3964 3963 3962 3961 3960 3959 3958 3957 3956 3955 3954 3953 3952 3951 3950 3949 3948 3947 3946 3945 3944 3943 3942 3941 3940 3939 3938 3937 3936 3935 3934 3933 3932 3931 3930 3929 3928 3927 3926 3925 3924 3923 3922 3921 3920 3919 3918 3917 3916 3915 3914 3913 3912 3911 3910 3909 3908 3907 3906 3905 3904 3903 3902 3901 3900 3899 3898 3897 3896 3895 3894 3893 3892 3891 3890 3889 3888 3887 3886 3885 3884 3883 3882 3881 3880 3879 3878 3877 3876 3875 3874 3873 3872 3871 3870 3869 3868 3867 3866 3865 3864 3863 3862 3861 3860 3859 3858 3857 3856 3855 3854 3853 3852 3851 3850 3849 3848 3847 3846 3845 3844 3843 3842 3841 3840 3839 3838 3837 3836 3835 3834 3833 3832 3831 3830 3829 3828 3827 3826 3825 3824 3823 3822 3821 3820 3819 3818 3817 3816 3815 3814 3813 3812 3811 3810 3809 3808 3807 3806 3805 3804 3803 3802 3801 3800 3799 3798 3797 3796 3795 3794 3793 3792 3791 3790 3789 3788 3787 3786 3785 3784 3783 3782 3781 3780 3779 3778 3777 3776 3775 3774 3773 3772 3771 3770 3769 3768 3767 3766 3765 3764 3763 3762 3761 3760 3759 3758 3757 3756 3755 3754 3753 3752 3751 3750 3749 3748 3747 3746 3745 3744 3743 3742 3741 3740 3739 3738 3737 3736 3735 3734 3733 3732 3731 3730 3729 3728 3727 3726 3725 3724 3723

3722 3721 3720 3719 3718 3717 3716 3715 3714 3713 3712 3711 3710 3709 3708 3707
3706 3705 3704 3703 3702 3701 3700 3699 3698 3697 3696 3695 3694 3693 3692 3691
3690 3689 3688 3687 3686 3685 3684 3683 3682 3681 3680 3679 3678 3677 3676 3675
3674 3673 3672 3671 3670 3669 3668 3667 3666 3665 3664 3663 3662 3661 3660 3659
3658 3657 3656 3655 3654 3653 3652 3651 3650 3649 3648 3647 3646 3645 3644 3643
3642 3641 3640 3639 3638 3637 3636 3635 3634 3633 3632 3631 3630 3629 3628 3627
3626 3625 3624 3623 3622 3621 3620 3619 3618 3617 3616 3615 3614 3613 3612 3611
3610 3609 3608 3607 3606 3605 3604 3603 3602 3601 3600 3599 3598 3597 3596 3595
3594 3593 3592 3591 3590 3589 3588 3587 3586 3585 3584 3583 3582 3581 3580 3579
3578 3577 3576 3575 3574 3573 3572 3571 3570 3569 3568 3567 3566 3565 3564 3563
3562 3561 3560 3559 3558 3557 3556 3555 3554 3553 3552 3551 3550 3549 3548 3547
3546 3545 3544 3543 3542 3541 3540 3539 3538 3537 3536 3535 3534 3533 3532 3531
3530 3529 3528 3527 3526 3525 3524 3523 3522 3521 3520 3519 3518 3517 3516 3515
3514 3513 3512 3511 3510 3509 3508 3507 3506 3505 3504 3503 3502 3501 3500 3499
3498 3497 3496 3495 3494 3493 3492 3491 3490 3489 3488 3487 3486 3485 3484 3483
3482 3481 3480 3479 3478 3477 3476 3475 3474 3473 3472 3471 3470 3469 3468 3467
3466 3465 3464 3463 3462 3461 3460 3459 3458 3457 3456 3455 3454 3453 3452 3451
3450 3449 3448 3447 3446 3445 3444 3443 3442 3441 3440 3439 3438 3437 3436 3435
3434 3433 3432 3431 3430 3429 3428 3427 3426 3425 3424 3423 3422 3421 3420 3419
3418 3417 3416 3415 3414 3413 3412 3411 3410 3409 3408 3407 3406 3405 3404 3403
3402 3401 3400 3399 3398 3397 3396 3395 3394 3393 3392 3391 3390 3389 3388 3387
3386 3385 3384 3383 3382 3381 3380 3379 3378 3377 3376 3375 3374 3373 3372 3371
3370 3369 3368 3367 3366 3365 3364 3363 3362 3361 3360 3359 3358 3357 3356 3355
3354 3353 3352 3351 3350 3349 3348 3347 3346 3345 3344 3343 3342 3341 3340 3339
3338 3337 3336 3335 3334 3333 3332 3331 3330 3329 3328 3327 3326 3325 3324 3323
3322 3321 3320 3319 3318 3317 3316 3315 3314 3313 3312 3311 3310 3309 3308 3307
3306 3305 3304 3303 3302 3301 3300 3299 3298 3297 3296 3295 3294 3293 3292 3291
3290 3289 3288 3287 3286 3285 3284 3283 3282 3281 3280 3279 3278 3277 3276 3275
3274 3273 3272 3271 3270 3269 3268 3267 3266 3265 3264 3263 3262 3261 3260 3259
3258 3257 3256 3255 3254 3253 3252 3251 3250 3249 3248 3247 3246 3245 3244 3243
3242 3241 3240 3239 3238 3237 3236 3235 3234 3233 3232 3231 3230 3229 3228 3227
3226 3225 3224 3223 3222 3221 3220 3219 3218 3217 3216 3215 3214 3213 3212 3211
3210 3209 3208 3207 3206 3205 3204 3203 3202 3201 3200 3199 3198 3197 3196 3195
3194 3193 3192 3191 3190 3189 3188 3187 3186 3185 3184 3183 3182 3181 3180 3179
3178 3177 3176 3175 3174 3173 3172 3171 3170 3169 3168 3167 3166 3165 3164 3163
3162 3161 3160 3159 3158 3157 3156 3155 3154 3153 3152 3151 3150 3149 3148 3147

3146 3145 3144 3143 3142 3141 3140 3139 3138 3137 3136 3135 3134 3133 3132 3131
3130 3129 3128 3127 3126 3125 3124 3123 3122 3121 3120 3119 3118 3117 3116 3115
3114 3113 3112 3111 3110 3109 3108 3107 3106 3105 3104 3103 3102 3101 3100 3099
3098 3097 3096 3095 3094 3093 3092 3091 3090 3089 3088 3087 3086 3085 3084 3083
3082 3081 3080 3079 3078 3077 3076 3075 3074 3073 3072 3071 3070 3069 3068 3067
3066 3065 3064 3063 3062 3061 3060 3059 3058 3057 3056 3055 3054 3053 3052 3051
3050 3049 3048 3047 3046 3045 3044 3043 3042 3041 3040 3039 3038 3037 3036 3035
3034 3033 3032 3031 3030 3029 3028 3027 3026 3025 3024 3023 3022 3021 3020 3019
3018 3017 3016 3015 3014 3013 3012 3011 3010 3009 3008 3007 3006 3005 3004 3003
3002 3001 3000 2999 2998 2997 2996 2995 2994 2993 2992 2991 2990 2989 2988 2987
2986 2985 2984 2983 2982 2981 2980 2979 2978 2977 2976 2975 2974 2973 2972 2971
2970 2969 2968 2967 2966 2965 2964 2963 2962 2961 2960 2959 2958 2957 2956 2955
2954 2953 2952 2951 2950 2949 2948 2947 2946 2945 2944 2943 2942 2941 2940 2939
2938 2937 2936 2935 2934 2933 2932 2931 2930 2929 2928 2927 2926 2925 2924 2923
2922 2921 2920 2919 2918 2917 2916 2915 2914 2913 2912 2911 2910 2909 2908 2907
2906 2905 2904 2903 2902 2901 2900 2899 2898 2897 2896 2895 2894 2893 2892 2891
2890 2889 2888 2887 2886 2885 2884 2883 2882 2881 2880 2879 2878 2877 2876 2875
2874 2873 2872 2871 2870 2869 2868 2867 2866 2865 2864 2863 2862 2861 2860 2859
2858 2857 2856 2855 2854 2853 2852 2851 2850 2849 2848 2847 2846 2845 2844 2843
2842 2841 2840 2839 2838 2837 2836 2835 2834 2833 2832 2831 2830 2829 2828 2827
2826 2825 2824 2823 2822 2821 2820 2819 2818 2817 2816 2815 2814 2813 2812 2811
2810 2809 2808 2807 2806 2805 2804 2803 2802 2801 2800 2799 2798 2797 2796 2795
2794 2793 2792 2791 2790 2789 2788 2787 2786 2785 2784 2783 2782 2781 2780 2779
2778 2777 2776 2775 2774 2773 2772 2771 2770 2769 2768 2767 2766 2765 2764 2763
2762 2761 2760 2759 2758 2757 2756 2755 2754 2753 2752 2751 2750 2749 2748 2747
2746 2745 2744 2743 2742 2741 2740 2739 2738 2737 2736 2735 2734 2733 2732 2731
2730 2729 2728 2727 2726 2725 2724 2723 2722 2721 2720 2719 2718 2717 2716 2715
2714 2713 2712 2711 2710 2709 2708 2707 2706 2705 2704 2703 2702 2701 2700 2699
2698 2697 2696 2695 2694 2693 2692 2691 2690 2689 2688 2687 2686 2685 2684 2683
2682 2681 2680 2679 2678 2677 2676 2675 2674 2673 2672 2671 2670 2669 2668 2667
2666 2665 2664 2663 2662 2661 2660 2659 2658 2657 2656 2655 2654 2653 2652 2651
2650 2649 2648 2647 2646 2645 2644 2643 2642 2641 2640 2639 2638 2637 2636 2635
2634 2633 2632 2631 2630 2629 2628 2627 2626 2625 2624 2623 2622 2621 2620 2619
2618 2617 2616 2615 2614 2613 2612 2611 2610 2609 2608 2607 2606 2605 2604 2603
2602 2601 2600 2599 2598 2597 2595 2594 2593 2592 2591 2590 2589 2588 2587 2586
2585 2584 2583 2582 2581 2580 2579 2578 2577 2576 2575 2574 2573 2572 2571 2570

2569 2568 2567 2566 2565 2564 2563 2562 2561 2560 2559 2558 2557 2556 2555 2554
2553 2552 2551 2550 2549 2548 2547 2546 2545 2544 2543 2542 2541 2540 2539 2538
2537 2536 2535 2534 2533 2532 2531 2530 2529 2528 2527 2526 2525 2524 2523 2522
2521 2520 2519 2518 2517 2516 2515 2514 2513 2512 2511 2510 2509 2508 2507 2506
2505 2504 2503 2502 2501 2500 2499 2498 2497 2496 2495 2494 2493 2492 2491 2490
2489 2488 2487 2486 2485 2484 2483 2482 2481 2480 2479 2478 2477 2476 2475 2474
2473 2472 2471 2470 2469 2468 2467 2466 2465 2464 2463 2462 2461 2460 2459 2458
2457 2456 2455 2454 2453 2452 2451 2450 2449 2448 2447 2446 2445 2444 2443 2442
2441 2440 2439 2438 2437 2436 2435 2434 2433 2432 2431 2430 2429 2428 2427 2426
2425 2424 2423 2422 2421 2420 2419 2418 2417 2416 2415 2414 2413 2412 2411 2410
2409 2408 2407 2406 2405 2404 2403 2402 2401 2400 2399 2398 2397 2396 2395 2394
2393 2392 2391 2390 2389 2388 2387 2386 2385 2384 2383 2382 2381 2380 2379 2378
2377 2376 2375 2374 2373 2372 2371 2370 2369 2368 2367 2366 2365 2364 2363 2362
2361 2360 2359 2358 2357 2356 2355 2354 2353 2352 2351 2350 2349 2348 2347 2346
2345 2344 2343 2342 2341 2340 2339 2338 2337 2336 2335 2334 2333 2332 2331 2330
2329 2328 2327 2326 2325 2324 2323 2322 2321 2320 2319 2318 2317 2316 2315 2314
2313 2312 2311 2310 2309 2308 2307 2306 2305 2304 2303 2302 2301 2300 2299 2298
2297 2296 2295 2294 2293 2292 2291 2290 2289 2288 2287 2286 2285 2284 2283 2282
2281 2280 2279 2278 2277 2276 2275 2274 2273 2272 2271 2270 2269 2268 2267 2266
2265 2264 2263 2262 2261 2260 2259 2258 2257 2256 2255 2254 2253 2252 2251 2250
2249 2248 2247 2246 2245 2244 2243 2242 2241 2240 2239 2238 2237 2236 2235 2234
2233 2232 2231 2230 2229 2228 2227 2226 2225 2224 2223 2222 2221 2220 2219 2218
2217 2216 2215 2214 2213 2212 2211 2210 2209 2208 2207 2206 2205 2204 2203 2202
2201 2200 2199 2198 2197 2196 2195 2194 2193 2192 2191 2190 2189 2188 2187 2186
2185 2184 2183 2182 2181 2180 2179 2178 2177 2176 2175 2174 2173 2172 2171 2170
2169 2168 2167 2166 2165 2164 2163 2162 2161 2160 2159 2158 2157 2156 2155 2154
2153 2152 2151 2150 2149 2148 2147 2146 2145 2144 2143 2142 2141 2140 2139 2138
2137 2136 2135 2134 2133 2132 2131 2130 2129 2128 2127 2126 2125 2124 2123 2122
2121 2120 2119 2118 2117 2116 2115 2114 2113 2112 2111 2110 2109 2108 2107 2106
2105 2104 2103 2102 2101 2100 2099 2098 2097 2096 2095 2094 2093 2092 2091 2090
2089 2088 2087 2086 2085 2084 2083 2082 2081 2080 2079 2078 2077 2076 2075 2074
2073 2072 2071 2070 2069 2068 2067 2066 2065 2064 2063 2062 2061 2060 2059 2058
2057 2056 2055 2054 2053 2052 2051 2050 2049 2048 2047 2046 2045 2044 2043 2042
2041 2040 2039 2038 2037 2036 2035 2034 2033 2032 2031 2030 2029 2028 2027 2026
2025 2024 2023 2022 2021 2020 2019 2018 2017 2016 2015 2014 2013 2012 2011 2010
2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1998 1997 1996 1995 1994

1993 1992 1991 1990 1989 1988 1987 1986 1985 1984 1983 1982 1981 1980 1979 1978
1977 1976 1975 1974 1973 1972 1971 1970 1969 1968 1967 1966 1965 1964 1963 1962
1961 1960 1959 1958 1957 1956 1955 1954 1953 1952 1951 1950 1949 1948 1947 1946
1945 1944 1943 1942 1941 1940 1939 1938 1937 1936 1935 1934 1933 1932 1931 1930
1929 1928 1927 1926 1925 1924 1923 1922 1921 1920 1919 1918 1917 1916 1915 1914
1913 1912 1911 1910 1909 1908 1907 1906 1905 1904 1903 1902 1901 1900 1899 1898
1897 1896 1895 1894 1893 1892 1891 1890 1889 1888 1887 1886 1885 1884 1883 1882
1881 1880 1879 1878 1877 1876 1875 1874 1873 1872 1871 1870 1869 1868 1867 1866
1865 1864 1863 1862 1861 1860 1859 1858 1857 1856 1855 1854 1853 1852 1851 1850
1849 1848 1847 1846 1845 1844 1843 1842 1841 1840 1839 1838 1837 1836 1835 1834
1833 1832 1831 1830 1829 1828 1827 1826 1825 1824 1823 1822 1821 1820 1819 1818
1817 1816 1815 1814 1813 1812 1811 1810 1809 1808 1807 1806 1805 1804 1803 1802
1801 1800 1799 1798 1797 1796 1795 1794 1793 1792 1791 1790 1789 1788 1787 1786
1785 1784 1783 1782 1781 1780 1779 1778 1777 1776 1775 1774 1773 1772 1771 1770
1769 1768 1767 1766 1765 1764 1763 1762 1761 1760 1759 1758 1757 1756 1755 1754
1753 1752 1751 1750 1749 1748 1747 1746 1745 1744 1743 1742 1741 1740 1739 1738
1737 1736 1735 1734 1733 1732 1731 1730 1729 1728 1727 1726 1725 1724 1723 1722
1721 1720 1719 1718 1717 1716 1715 1714 1713 1712 1711 1710 1709 1708 1707 1706
1705 1704 1703 1702 1701 1700 1699 1698 1697 1696 1695 1694 1693 1692 1691 1690
1689 1688 1687 1686 1685 1684 1683 1682 1681 1680 1679 1678 1677 1676 1675 1674
1673 1672 1671 1670 1669 1668 1667 1666 1665 1664 1663 1662 1661 1660 1659 1658
1657 1656 1655 1654 1653 1652 1651 1650 1649 1648 1647 1646 1645 1644 1643 1642
1641 1640 1639 1638 1637 1636 1635 1634 1633 1632 1631 1630 1629 1628 1627 1626
1625 1624 1623 1622 1621 1620 1619 1618 1617 1616 1615 1614 1613 1612 1611 1610
1609 1608 1607 1606 1605 1604 1603 1602 1601 1600 1599 1598 1597 1596 1595 1594
1593 1592 1591 1590 1589 1588 1587 1586 1585 1584 1583 1582 1581 1580 1579 1578
1577 1576 1575 1574 1573 1572 1571 1570 1569 1568 1567 1566 1565 1564 1563 1562
1561 1560 1559 1558 1557 1556 1555 1554 1553 1552 1551 1550 1549 1548 1547 1546
1545 1544 1543 1542 1541 1540 1539 1538 1537 1536 1535 1534 1533 1532 1531 1530
1529 1528 1527 1526 1525 1524 1523 1522 1521 1520 1519 1518 1517 1516 1515 1514
1513 1512 1511 1510 1509 1508 1507 1506 1505 1504 1503 1502 1501 1500 1499 1498
1497 1496 1495 1494 1493 1492 1491 1490 1489 1488 1487 1486 1485 1484 1483 1482
1481 1480 1479 1478 1477 1476 1475 1474 1473 1472 1471 1470 1469 1468 1467 1466
1465 1464 1463 1462 1461 1460 1459 1458 1457 1456 1455 1454 1453 1452 1451 1450
1449 1448 1447 1446 1445 1444 1443 1442 1441 1440 1439 1438 1437 1436 1435 1434
1433 1432 1431 1430 1429 1428 1427 1426 1425 1424 1423 1422 1421 1420 1419 1418

1417 1416 1415 1414 1413 1412 1411 1410 1409 1408 1407 1406 1405 1404 1403 1402
1401 1400 1399 1398 1397 1396 1395 1394 1393 1392 1391 1390 1389 1388 1387 1386
1385 1384 1383 1382 1381 1380 1379 1378 1377 1376 1375 1374 1373 1372 1371 1370
1369 1368 1367 1366 1365 1364 1363 1362 1361 1360 1359 1358 1357 1356 1355 1354
1353 1352 1351 1350 1349 1348 1347 1346 1345 1344 1343 1342 1341 1340 1339 1338
1337 1336 1335 1334 1333 1332 1331 1330 1329 1328 1327 1326 1325 1324 1323 1322
1321 1320 1319 1318 1317 1316 1315 1314 1313 1312 1311 1310 1309 1308 1307 1306
1305 1304 1303 1302 1301 1300 1299 1298 1297 1296 1295 1294 1293 1292 1291 1290
1289 1288 1287 1286 1285 1284 1283 1282 1281 1280 1279 1278 1277 1276 1275 1274
1273 1272 1271 1270 1269 1268 1267 1266 1265 1264 1263 1262 1261 1260 1259 1258
1257 1256 1255 1254 1253 1252 1251 1250 1249 1248 1247 1246 1245 1244 1243 1242
1241 1240 1239 1238 1237 1236 1235 1234 1233 1232 1231 1230 1229 1228 1227 1226
1225 1224 1223 1222 1221 1220 1219 1218 1217 1216 1215 1214 1213 1212 1211 1210
1209 1208 1207 1206 1205 1204 1203 1202 1201 1200 1199 1198 1197 1196 1195 1194
1193 1192 1191 1190 1189 1188 1187 1186 1185 1184 1183 1182 1181 1180 1179 1178
1177 1176 1175 1174 1173 1172 1171 1170 1169 1168 1167 1166 1165 1164 1163 1162
1161 1160 1159 1158 1157 1156 1155 1154 1153 1152 1151 1150 1149 1148 1147 1146
1145 1144 1143 1142 1141 1140 1139 1138 1137 1136 1135 1134 1133 1132 1131 1130
1129 1128 1127 1126 1125 1124 1123 1122 1121 1120 1119 1118 1117 1116 1115 1114
1113 1112 1111 1110 1109 1108 1107 1106 1105 1104 1103 1102 1101 1100 1099 1098
1097 1096 1095 1094 1093 1092 1091 1090 1089 1088 1087 1086 1085 1084 1083 1082
1081 1080 1079 1078 1077 1076 1075 1074 1073 1072 1071 1070 1069 1068 1067 1066
1065 1064 1063 1062 1061 1060 1059 1058 1057 1056 1055 1054 1053 1052 1051 1050
1049 1048 1047 1046 1045 1044 1043 1042 1041 1040 1039 1038 1037 1036 1035 1034
1033 1032 1030 1029 1028 1027 1026 1025 1024 1023 1022 1021 1020 1019 1018 1017
1016 1015 1014 1013 1012 1011 1010 1009 1008 1007 1006 1005 1004 1003 1002 1001
1000 999 998 997 996 995 994 993 992 991 990 989 988 987 986 985 984 983 982 981
980 979 978 977 976 975 974 973 972 971 970 969 968 967 966 965 964 963 962 961
960 959 958 957 956 955 954 953 952 951 950 949 948 947 946 945 944 943 942 941
940 939 938 937 936 935 934 933 932 931 930 929 928 927 926 925 924 923 922 921
920 919 918 917 916 915 914 913 912 911 910 909 908 907 906 905 904 903 902 901
900 899 898 897 896 895 894 893 892 891 890 889 888 887 886 885 884 883 882 881
880 879 878 877 876 875 874 873 872 871 870 869 868 867 866 865 864 863 862 861
860 859 858 857 856 855 854 853 852 851 850 849 848 847 846 845 844 843 842 841
840 839 838 837 836 835 834 833 832 831 830 829 828 827 826 825 824 823 822 821
820 819 818 817 816 815 814 813 812 811 810 809 808 807 806 805 804 803 802 801

800 799 798 797 796 795 794 793 792 791 790 789 788 787 786 785 784 783 782 781
780 779 778 777 776 775 774 773 772 771 770 769 768 767 766 765 764 763 762 761
760 759 758 757 756 755 754 753 752 751 750 749 748 747 746 745 744 743 742 741
740 739 738 737 736 735 734 733 732 731 730 729 728 727 726 725 724 723 722 721
720 719 718 717 716 715 714 713 712 711 710 709 708 707 706 705 704 703 702 701
700 699 698 697 696 695 694 693 692 691 690 689 688 687 686 685 684 683 682 681
680 679 678 677 676 675 674 673 672 671 670 669 668 667 666 665 664 663 662 661
660 659 658 657 656 655 654 653 652 651 650 649 648 647 646 645 644 643 642 641
640 639 638 637 636 635 634 633 632 631 630 629 628 627 626 625 624 623 622 621
620 619 618 617 616 615 614 613 612 611 610 609 608 607 606 605 604 603 602 601
600 599 598 597 596 595 594 593 592 591 590 589 588 587 586 585 584 583 582 581
580 579 578 577 576 575 574 573 572 571 570 569 568 567 566 565 564 563 562 560
559 558 557 556 555 554 553 552 551 550 549 548 547 546 545 544 543 542 541 540
539 538 537 536 535 534 533 532 531 530 529 528 527 526 525 524 523 522 521 520
519 518 517 516 515 514 513 512 511 510 509 508 507 506 505 504 503 502 501 500
499 498 497 496 495 494 493 492 491 490 489 488 487 486 485 484 483 482 481 480
479 478 477 476 475 474 473 472 471 470 469 468 467 466 465 464 463 462 461 460
459 458 457 456 455 454 453 452 451 450 449 448 447 446 445 444 443 442 441 440
439 438 437 436 435 434 433 432 431 430 429 428 427 426 425 424 423 422 421 420
419 418 417 416 415 414 413 412 411 410 409 408 407 406 405 404 403 402 401 400
399 398 397 396 395 394 393 392 391 390 389 388 387 386 385 384 383 382 381 380
379 378 377 376 375 374 373 372 371 370 369 368 367 366 365 364 363 362 361 360
359 358 357 356 355 354 353 352 351 350 349 348 347 346 345 344 343 342 341 340
339 338 337 336 335 334 333 332 331 330 329 328 327 326 325 324 323 322 321 320
319 318 317 316 315 314 313 312 311 310 309 308 307 306 305 304 303 302 301 300
299 298 297 296 295 294 293 292 291 290 289 288 287 286 285 284 283 282 281 280
279 278 277 276 275 274 273 272 271 270 269 268 267 266 265 264 263 262 261 260
259 258 257 256 255 254 253 252 251 250 249 248 247 246 245 244 243 242 241 240
239 238 237 236 235 234 233 232 231 230 229 228 227 226 225 224 223 222 221 220
219 218 217 216 215 214 213 212 211 210 209 208 207 206 205 204 203 202 201 200
199 198 197 196 195 194 193 192 191 190 189 188 187 186 185 184 183 182 181 180
179 178 177 176 175 174 173 172 171 170 169 168 167 166 165 164 163 162 161 160
159 158 157 156 155 154 153 152 151 150 149 148 147 146 145 144 143 142 141 140
139 138 137 136 135 134 133 132 131 130 129 128 127 126 125 124 123 122 121 120
119 118 117 116 115 114 113 112 111 110 109 108 107 106 105 104 103 102 101 100
99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72

71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

attributes:

2 3 9 17 18 20 24 26 28 43 44 49 51 54 55 56

objects:

4026 4025 4024 4023 4022 4021 4020 4019 4018 4017 4016 4015 4014 4013 4012 4011 4010 4009 4008 4007 4006 4005 4004 4003 4002 4001 4000 3999 3998 3997 3996 3995 3994 3993 3992 3991 3990 3989 3988 3987 3986 3985 3984 3983 3982 3981 3980 3979 3978 3977 3976 3975 3974 3973 3972 3971 3970 3969 3968 3967 3966 3965 3964 3963 3962 3961 3960 3959 3958 3957 3956 3955 3954 3953 3952 3951 3950 3949 3948 3947 3946 3945 3944 3943 3942 3941 3940 3939 3938 3937 3936 3935 3934 3933 3932 3931 3930 3929 3928 3927 3926 3925 3924 3923 3922 3921 3920 3919 3918 3917 3916 3915 3914 3913 3912 3911 3910 3909 3908 3907 3906 3905 3904 3903 3902 3901 3900 3899 3898 3897 3896 3895 3894 3893 3892 3891 3890 3889 3888 3887 3886 3885 3884 3883 3882 3881 3880 3879 3878 3877 3876 3875 3874 3873 3872 3871 3870 3869 3868 3867 3866 3865 3864 3863 3862 3861 3860 3859 3858 3857 3856 3855 3854 3853 3852 3851 3850 3849 3848 3847 3846 3845 3844 3843 3842 3841 3840 3839 3838 3837 3836 3835 3834 3833 3832 3831 3830 3829 3828 3827 3826 3825 3824 3823 3822 3821 3820 3819 3818 3817 3816 3815 3814 3813 3812 3811 3810 3809 3808 3807 3806 3805 3804 3803 3802 3801 3800 3799 3798 3797 3796 3795 3794 3793 3792 3791 3790 3789 3788 3787 3786 3785 3784 3783 3782 3781 3780 3779 3778 3777 3776 3775 3774 3773 3772 3771 3770 3769 3768 3767 3766 3765 3764 3763 3762 3761 3760 3759 3758 3757 3756 3755 3754 3753 3752 3751 3750 3749 3748 3747 3746 3745 3744 3743 3742 3741 3740 3739 3738 3737 3736 3735 3734 3733 3732 3731 3730 3729 3728 3727 3726 3725 3724 3723 3722 3721 3720 3719 3718 3717 3716 3715 3714 3713 3712 3711 3710 3709 3708 3707 3706 3705 3704 3703 3702 3701 3700 3699 3698 3697 3696 3695 3694 3693 3692 3691 3690 3689 3688 3687 3686 3685 3684 3683 3682 3681 3680 3679 3678 3677 3676 3675 3674 3673 3672 3671 3670 3669 3668 3667 3666 3665 3664 3663 3662 3661 3660 3659 3658 3657 3656 3655 3654 3653 3652 3651 3650 3649 3648 3647 3646 3645 3644 3643 3642 3641 3640 3639 3638 3637 3636 3635 3634 3633 3632 3631 3630 3629 3628 3627 3626 3625 3624 3623 3622 3621 3620 3619 3618 3617 3616 3615 3614 3613 3612 3611 3610 3609 3608 3607 3606 3605 3604 3603 3602 3601 3600 3599 3598 3597 3596 3595 3594 3593 3592 3591 3590 3589 3588 3587 3586 3585 3584 3583 3582 3581 3580 3579 3578 3577 3576 3575 3574 3573 3572 3571 3570 3569 3568 3567 3566 3565 3564 3563 3562 3561 3560 3559 3558 3557 3556 3555 3554 3553 3552 3551 3550 3549 3548 3547

3546 3545 3544 3543 3542 3541 3540 3539 3538 3537 3536 3535 3534 3533 3532 3531
3530 3529 3528 3527 3526 3525 3524 3523 3522 3521 3520 3519 3518 3517 3516 3515
3514 3513 3512 3511 3510 3509 3508 3507 3506 3505 3504 3503 3502 3501 3500 3499
3498 3497 3496 3495 3494 3493 3492 3491 3490 3489 3488 3487 3486 3485 3484 3483
3482 3481 3480 3479 3478 3477 3476 3475 3474 3473 3472 3471 3470 3469 3468 3467
3466 3465 3464 3463 3462 3461 3460 3459 3458 3457 3456 3455 3454 3453 3452 3451
3450 3449 3448 3447 3446 3445 3444 3443 3442 3441 3440 3439 3438 3437 3436 3435
3434 3433 3432 3431 3430 3429 3428 3427 3426 3425 3424 3423 3422 3421 3420 3419
3418 3417 3416 3415 3414 3413 3412 3411 3410 3409 3408 3407 3406 3405 3404 3403
3402 3401 3400 3399 3398 3397 3396 3395 3394 3393 3392 3391 3390 3389 3388 3387
3386 3385 3384 3383 3382 3381 3380 3379 3378 3377 3376 3375 3374 3373 3372 3371
3370 3369 3368 3367 3366 3365 3364 3363 3362 3361 3360 3359 3358 3357 3356 3355
3354 3353 3352 3351 3350 3349 3348 3347 3346 3345 3344 3343 3342 3341 3340 3339
3338 3337 3336 3335 3334 3333 3332 3331 3330 3329 3328 3327 3326 3325 3324 3323
3322 3321 3320 3319 3318 3317 3316 3315 3314 3313 3312 3311 3310 3309 3308 3307
3306 3305 3304 3303 3302 3301 3300 3299 3298 3297 3296 3295 3294 3293 3292 3291
3290 3289 3288 3287 3286 3285 3284 3283 3282 3281 3280 3279 3278 3277 3276 3275
3274 3273 3272 3271 3270 3269 3268 3267 3266 3265 3264 3263 3262 3261 3260 3259
3258 3257 3256 3255 3254 3253 3252 3251 3250 3249 3248 3247 3246 3245 3244 3243
3242 3241 3240 3239 3238 3237 3236 3235 3234 3233 3232 3231 3230 3229 3228 3227
3226 3225 3224 3223 3222 3221 3220 3219 3218 3217 3216 3215 3214 3213 3212 3211
3210 3209 3208 3207 3206 3205 3204 3203 3202 3201 3200 3199 3198 3197 3196 3195
3194 3193 3192 3191 3190 3189 3188 3187 3186 3185 3184 3183 3182 3181 3180 3179
3178 3177 3176 3175 3174 3173 3172 3171 3170 3169 3168 3167 3166 3165 3164 3163
3162 3161 3160 3159 3158 3157 3156 3155 3154 3153 3152 3151 3150 3149 3148 3147
3146 3145 3144 3143 3142 3141 3140 3139 3138 3137 3136 3135 3134 3133 3132 3131
3130 3129 3128 3127 3126 3125 3124 3123 3122 3121 3120 3119 3118 3117 3116 3115
3114 3113 3112 3111 3110 3109 3108 3107 3106 3105 3104 3103 3102 3101 3100 3099
3098 3097 3096 3095 3094 3093 3092 3091 3090 3089 3088 3087 3086 3085 3084 3083
3082 3081 3080 3079 3078 3077 3076 3075 3074 3073 3072 3071 3070 3069 3068 3067
3066 3065 3064 3063 3062 3061 3060 3059 3058 3057 3056 3055 3054 3053 3052 3051
3050 3049 3048 3047 3046 3045 3044 3043 3042 3041 3040 3039 3038 3037 3036 3035
3034 3033 3032 3031 3030 3029 3028 3027 3026 3025 3024 3023 3022 3021 3020 3019
3018 3017 3016 3015 3014 3013 3012 3011 3010 3009 3008 3007 3006 3005 3004 3003
3002 3001 3000 2999 2998 2997 2996 2995 2994 2993 2992 2991 2990 2989 2988 2987
2986 2985 2984 2983 2982 2981 2980 2979 2978 2977 2976 2975 2974 2973 2972 2971

2970 2969 2968 2967 2966 2965 2964 2963 2962 2961 2960 2959 2958 2957 2956 2955
2954 2953 2952 2951 2950 2949 2948 2947 2946 2945 2944 2943 2942 2941 2940 2939
2938 2937 2936 2935 2934 2933 2932 2931 2930 2929 2928 2927 2926 2925 2924 2923
2922 2921 2920 2919 2918 2917 2916 2915 2914 2913 2912 2911 2910 2909 2908 2907
2906 2905 2904 2903 2902 2901 2900 2899 2898 2897 2896 2895 2894 2893 2892 2891
2890 2889 2888 2887 2886 2885 2884 2883 2882 2881 2880 2879 2878 2877 2876 2875
2874 2873 2872 2871 2870 2869 2868 2867 2866 2865 2864 2863 2862 2861 2860 2859
2858 2857 2856 2855 2854 2853 2852 2851 2850 2849 2848 2847 2846 2845 2844 2843
2842 2841 2840 2839 2838 2837 2836 2835 2834 2833 2832 2831 2830 2829 2828 2827
2826 2825 2824 2823 2822 2821 2820 2819 2818 2817 2816 2815 2814 2813 2812 2811
2810 2809 2808 2807 2806 2805 2804 2803 2802 2801 2800 2799 2798 2797 2796 2795
2794 2793 2792 2791 2790 2789 2788 2787 2786 2785 2784 2783 2782 2781 2780 2779
2778 2777 2776 2775 2774 2773 2772 2771 2770 2769 2768 2767 2766 2765 2764 2763
2762 2761 2760 2759 2758 2757 2756 2755 2754 2753 2752 2751 2750 2749 2748 2747
2746 2745 2744 2743 2742 2741 2740 2739 2738 2737 2736 2735 2734 2733 2732 2731
2730 2729 2728 2727 2726 2725 2724 2723 2722 2721 2720 2719 2718 2717 2716 2715
2714 2713 2712 2711 2710 2709 2708 2707 2706 2705 2704 2703 2702 2701 2700 2699
2698 2697 2696 2695 2694 2693 2692 2691 2690 2689 2688 2687 2686 2685 2684 2683
2682 2681 2680 2679 2678 2677 2676 2675 2674 2673 2672 2671 2670 2669 2668 2667
2666 2665 2664 2663 2662 2661 2660 2659 2658 2657 2656 2655 2654 2653 2652 2651
2650 2649 2648 2647 2646 2645 2644 2643 2642 2641 2640 2639 2638 2637 2636 2635
2634 2633 2632 2631 2630 2629 2628 2627 2626 2625 2624 2623 2622 2621 2620 2619
2618 2617 2616 2615 2614 2613 2612 2611 2610 2609 2608 2607 2606 2605 2604 2603
2602 2601 2600 2599 2598 2597 2596 2595 2594 2593 2592 2591 2590 2589 2588 2587
2586 2585 2584 2583 2582 2581 2580 2579 2578 2577 2576 2575 2574 2573 2572 2571
2570 2569 2568 2567 2566 2565 2564 2563 2562 2561 2560 2559 2558 2557 2556 2555
2554 2553 2552 2551 2550 2549 2548 2547 2546 2545 2544 2543 2542 2541 2540 2539
2538 2537 2536 2535 2534 2533 2532 2531 2530 2529 2528 2527 2526 2525 2524 2523
2522 2521 2520 2519 2518 2517 2516 2515 2514 2513 2512 2511 2510 2509 2508 2507
2506 2505 2504 2503 2502 2501 2500 2499 2498 2497 2496 2495 2494 2493 2492 2491
2490 2489 2488 2487 2486 2485 2484 2483 2482 2481 2480 2479 2478 2477 2476 2475
2474 2473 2472 2471 2470 2469 2468 2467 2466 2465 2464 2463 2462 2461 2460 2459
2458 2457 2456 2455 2454 2453 2452 2451 2450 2449 2448 2447 2446 2445 2444 2443
2442 2441 2440 2439 2438 2437 2436 2435 2434 2433 2432 2431 2430 2429 2428 2427
2426 2425 2424 2423 2422 2421 2420 2419 2418 2417 2416 2415 2414 2413 2412 2411
2410 2409 2408 2407 2406 2405 2404 2403 2402 2401 2400 2399 2398 2397 2396 2395

2394 2393 2392 2391 2390 2389 2388 2387 2386 2385 2384 2383 2382 2381 2380 2379
2378 2377 2376 2375 2374 2373 2372 2371 2370 2369 2368 2367 2366 2365 2364 2363
2362 2361 2360 2359 2358 2357 2356 2355 2354 2353 2352 2351 2350 2349 2348 2347
2346 2345 2344 2343 2342 2341 2340 2339 2338 2337 2336 2335 2334 2333 2332 2331
2330 2329 2328 2327 2326 2325 2324 2323 2322 2321 2320 2319 2318 2317 2316 2315
2314 2313 2312 2311 2310 2309 2308 2307 2306 2305 2304 2303 2302 2301 2300 2299
2298 2297 2296 2295 2294 2293 2292 2291 2290 2289 2288 2287 2286 2285 2284 2283
2282 2281 2280 2279 2278 2277 2276 2275 2274 2273 2272 2271 2270 2269 2268 2267
2266 2265 2264 2263 2262 2261 2260 2259 2258 2257 2256 2255 2254 2253 2252 2251
2250 2249 2248 2247 2246 2245 2244 2243 2242 2241 2240 2239 2238 2237 2236 2235
2234 2233 2232 2231 2230 2229 2228 2227 2226 2225 2224 2223 2222 2221 2220 2219
2218 2217 2216 2215 2214 2213 2212 2211 2210 2209 2208 2207 2206 2205 2204 2203
2202 2201 2200 2199 2198 2197 2196 2195 2194 2193 2192 2191 2190 2189 2188 2187
2186 2185 2184 2183 2182 2181 2180 2179 2178 2177 2176 2175 2174 2173 2172 2171
2170 2169 2168 2167 2166 2165 2164 2163 2162 2161 2160 2159 2158 2157 2156 2155
2154 2153 2152 2151 2150 2149 2148 2147 2146 2145 2144 2143 2142 2141 2140 2139
2138 2137 2136 2135 2134 2133 2132 2131 2130 2129 2128 2127 2126 2125 2124 2123
2122 2121 2120 2119 2118 2117 2116 2115 2114 2113 2112 2111 2110 2109 2108 2107
2106 2105 2104 2103 2102 2101 2100 2099 2098 2097 2096 2095 2094 2093 2092 2091
2090 2089 2088 2087 2086 2085 2084 2083 2082 2081 2080 2079 2078 2077 2076 2075
2074 2073 2072 2071 2070 2069 2068 2067 2066 2065 2064 2063 2062 2061 2060 2059
2058 2057 2056 2055 2054 2053 2052 2051 2050 2049 2048 2047 2046 2045 2044 2043
2042 2041 2040 2039 2038 2037 2036 2035 2034 2033 2032 2031 2030 2029 2028 2027
2026 2025 2024 2023 2022 2021 2020 2019 2018 2017 2016 2015 2014 2013 2012 2011
2010 2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1998 1997 1996 1995
1994 1993 1992 1991 1990 1989 1988 1987 1986 1985 1984 1983 1982 1981 1980 1979
1978 1977 1976 1975 1974 1973 1972 1971 1970 1969 1968 1967 1966 1965 1964 1963
1962 1961 1960 1959 1958 1957 1956 1955 1954 1953 1952 1951 1950 1949 1948 1947
1946 1945 1944 1943 1942 1941 1940 1939 1938 1937 1936 1935 1934 1933 1932 1931
1930 1929 1928 1927 1926 1925 1924 1923 1922 1921 1920 1919 1918 1917 1916 1915
1914 1913 1912 1911 1910 1909 1908 1907 1906 1905 1904 1903 1902 1901 1900 1899
1898 1897 1896 1895 1894 1893 1892 1891 1890 1889 1888 1887 1886 1885 1884 1883
1882 1881 1880 1879 1878 1877 1876 1875 1874 1873 1872 1871 1870 1869 1868 1867
1866 1865 1864 1863 1862 1861 1860 1859 1858 1857 1856 1855 1854 1853 1852 1851
1850 1849 1848 1847 1846 1845 1844 1843 1842 1841 1840 1839 1838 1837 1836 1835
1834 1833 1832 1831 1830 1829 1828 1827 1826 1825 1824 1823 1822 1821 1820 1819

1818 1817 1816 1815 1814 1813 1812 1811 1810 1809 1808 1807 1806 1805 1804 1803
1802 1801 1800 1799 1798 1797 1796 1795 1794 1793 1792 1791 1790 1789 1788 1787
1786 1785 1784 1783 1782 1781 1780 1779 1778 1777 1776 1775 1774 1773 1772 1771
1770 1769 1768 1767 1766 1765 1764 1763 1762 1761 1760 1759 1758 1757 1756 1755
1754 1753 1752 1751 1750 1749 1748 1747 1746 1745 1744 1743 1742 1741 1740 1739
1738 1737 1736 1735 1734 1733 1732 1731 1730 1729 1728 1727 1726 1725 1724 1723
1722 1721 1720 1719 1718 1717 1716 1715 1714 1713 1712 1711 1710 1709 1708 1707
1706 1705 1704 1703 1702 1701 1700 1699 1698 1697 1696 1695 1694 1693 1692 1691
1690 1689 1688 1687 1686 1685 1684 1683 1682 1681 1680 1679 1678 1677 1676 1675
1674 1673 1672 1671 1670 1669 1668 1667 1666 1665 1664 1663 1662 1661 1660 1659
1658 1657 1656 1655 1654 1653 1652 1651 1650 1649 1648 1647 1646 1645 1644 1643
1642 1641 1640 1639 1638 1637 1636 1635 1634 1633 1632 1631 1630 1629 1628 1627
1626 1625 1624 1623 1622 1621 1620 1619 1618 1617 1616 1615 1614 1613 1612 1611
1610 1609 1608 1607 1606 1605 1604 1603 1602 1601 1600 1599 1598 1597 1596 1595
1594 1593 1592 1591 1590 1589 1588 1587 1586 1585 1584 1583 1582 1581 1580 1579
1578 1577 1576 1575 1574 1573 1572 1571 1570 1569 1568 1567 1566 1565 1564 1563
1562 1561 1560 1559 1558 1557 1556 1555 1554 1553 1552 1551 1550 1549 1548 1547
1546 1545 1544 1543 1542 1541 1540 1539 1538 1537 1536 1535 1534 1533 1532 1531
1530 1529 1528 1527 1526 1525 1524 1523 1522 1521 1520 1519 1518 1517 1516 1515
1514 1513 1512 1511 1510 1509 1508 1507 1506 1505 1504 1503 1502 1501 1500 1499
1498 1497 1496 1495 1494 1493 1492 1491 1490 1489 1488 1487 1486 1485 1484 1483
1482 1481 1480 1479 1478 1477 1476 1475 1474 1473 1472 1471 1470 1469 1468 1467
1466 1465 1464 1463 1462 1461 1460 1459 1458 1457 1456 1455 1454 1453 1452 1451
1450 1449 1448 1447 1446 1445 1444 1443 1442 1441 1440 1439 1438 1437 1436 1435
1434 1433 1432 1431 1430 1429 1428 1427 1426 1425 1424 1423 1422 1421 1420 1419
1418 1417 1416 1415 1414 1413 1412 1411 1410 1409 1408 1407 1406 1405 1404 1403
1402 1401 1400 1399 1398 1397 1396 1395 1394 1393 1392 1391 1390 1389 1388 1387
1386 1385 1384 1383 1382 1381 1380 1379 1378 1377 1376 1375 1374 1373 1372 1371
1370 1369 1368 1367 1366 1365 1364 1363 1362 1361 1360 1359 1358 1357 1356 1355
1354 1353 1352 1351 1350 1349 1348 1347 1346 1345 1344 1343 1342 1341 1340 1339
1338 1337 1336 1335 1334 1333 1332 1331 1330 1329 1328 1327 1326 1325 1324 1323
1322 1321 1320 1319 1318 1317 1316 1315 1314 1313 1312 1311 1310 1309 1308 1307
1306 1305 1304 1303 1302 1301 1300 1299 1298 1297 1296 1295 1294 1293 1292 1291
1290 1289 1288 1287 1286 1285 1284 1283 1282 1281 1280 1279 1278 1277 1276 1275
1274 1273 1272 1271 1270 1269 1268 1267 1266 1265 1264 1263 1262 1261 1260 1259
1258 1257 1256 1255 1254 1253 1252 1251 1250 1249 1248 1247 1246 1245 1244 1243

1242 1241 1240 1239 1238 1237 1236 1235 1234 1233 1232 1231 1230 1229 1228 1227
1226 1225 1224 1223 1222 1221 1220 1219 1218 1217 1216 1215 1214 1213 1212 1211
1210 1209 1208 1207 1206 1205 1204 1203 1202 1201 1200 1199 1198 1197 1196 1195
1194 1193 1192 1191 1190 1189 1188 1187 1186 1185 1184 1183 1182 1181 1180 1179
1178 1177 1176 1175 1174 1173 1172 1171 1170 1169 1168 1167 1166 1165 1164 1163
1162 1161 1160 1159 1158 1157 1156 1155 1154 1153 1152 1151 1150 1149 1148 1147
1146 1145 1144 1143 1142 1141 1140 1139 1138 1137 1136 1135 1134 1133 1132 1131
1130 1129 1128 1127 1126 1125 1124 1123 1122 1121 1120 1119 1118 1117 1116 1115
1114 1113 1112 1111 1110 1109 1108 1107 1106 1105 1104 1103 1102 1101 1100 1099
1098 1097 1096 1095 1094 1093 1092 1091 1090 1089 1088 1087 1086 1085 1084 1083
1082 1081 1080 1079 1078 1077 1076 1075 1074 1073 1072 1071 1070 1069 1068 1067
1066 1065 1064 1063 1062 1061 1060 1059 1058 1057 1056 1055 1054 1053 1052 1051
1050 1049 1048 1047 1046 1045 1044 1043 1042 1041 1040 1039 1038 1037 1036 1035
1034 1033 1032 1030 1029 1028 1027 1026 1025 1024 1023 1022 1021 1020 1019 1018
1017 1016 1015 1014 1013 1012 1011 1010 1009 1008 1007 1006 1005 1004 1003 1002
1001 1000 999 998 997 996 995 994 993 992 991 990 989 988 987 986 985 984 983 982
981 980 979 978 977 976 975 974 973 972 971 970 969 968 967 966 965 964 963 962
961 960 959 958 957 956 955 954 953 952 951 950 949 948 947 946 945 944 943 942
941 940 939 938 937 936 935 934 933 932 931 930 929 928 927 926 925 924 923 922
921 920 919 918 917 916 915 914 913 912 911 910 909 908 907 906 905 904 903 902
901 900 899 898 897 896 895 894 893 892 891 890 889 888 887 886 885 884 883 882
881 880 879 878 877 876 875 874 873 872 871 870 869 868 867 866 865 864 863 862
861 860 859 858 857 856 855 854 853 852 851 850 849 848 847 846 845 844 843 842
841 840 839 838 837 836 835 834 833 832 831 830 829 828 827 826 825 824 823 822
821 820 819 818 817 816 815 814 813 812 811 810 809 808 807 806 805 804 803 802
801 800 799 798 797 796 795 794 793 792 791 790 789 788 787 786 785 784 783 782
781 780 779 778 777 776 775 774 773 772 771 770 769 768 767 766 765 764 763 762
761 760 759 758 757 756 755 754 753 752 751 750 749 748 747 746 745 744 743 742
741 740 739 738 737 736 735 734 733 732 731 730 729 728 727 726 725 724 723 722
721 720 719 718 717 716 715 714 713 712 711 710 709 708 707 706 705 704 703 702
701 700 699 698 697 696 695 694 693 692 691 690 689 688 687 686 685 684 683 682
681 680 679 678 677 676 675 674 673 672 671 670 669 668 667 666 665 664 663 662
661 660 659 658 657 656 655 654 653 652 651 650 649 648 647 646 645 644 643 642
641 640 639 638 637 636 635 634 633 632 631 630 629 628 627 626 625 624 623 622
621 620 619 618 617 616 615 614 613 612 611 610 609 608 607 606 605 604 603 602
601 600 599 598 597 596 595 594 593 592 591 590 589 588 587 586 585 584 583 582

581 580 579 578 577 576 575 574 573 572 571 570 569 568 567 566 565 564 563 562
561 560 559 558 557 556 555 554 553 552 551 550 549 548 547 546 545 544 543 542
541 540 539 538 537 536 535 534 533 532 531 530 529 528 527 526 525 524 523 522
521 520 519 518 517 516 515 514 513 512 511 510 509 508 507 506 505 504 503 502
501 500 499 498 497 496 495 494 493 492 491 490 489 488 487 486 485 484 483 482
481 480 479 478 477 476 475 474 473 472 471 470 469 468 467 466 465 464 463 462
461 460 459 458 457 456 455 454 453 452 451 450 449 448 447 446 445 444 443 442
441 440 439 438 437 436 435 434 433 432 431 430 429 428 427 426 425 424 423 422
421 420 419 418 417 416 415 414 413 412 411 410 409 408 407 406 405 404 403 402
401 400 399 398 397 396 395 394 393 392 391 390 389 388 387 386 385 384 383 382
381 380 379 378 377 376 375 374 373 372 371 370 369 368 367 366 365 364 363 362
361 360 359 358 357 356 355 354 353 352 351 350 349 348 347 346 345 344 343 342
341 340 339 338 337 336 335 334 333 332 331 330 329 328 327 326 325 324 323 322
321 320 319 318 317 316 315 314 313 312 311 310 309 308 307 306 305 304 303 302
301 300 299 298 297 296 295 294 293 292 291 290 289 288 287 286 285 284 283 282
281 280 279 278 277 276 275 274 273 272 271 270 269 268 267 266 265 264 263 262
261 260 259 258 257 256 255 254 253 252 251 250 249 248 247 246 245 244 243 242
241 240 239 238 237 236 235 234 233 232 231 230 229 228 227 226 225 224 223 222
221 220 219 218 217 216 215 214 213 212 211 210 209 208 207 206 205 204 203 202
201 200 199 198 197 196 195 194 193 192 191 190 189 188 187 186 185 184 183 182
181 180 179 178 177 176 175 174 173 172 171 170 169 168 167 166 165 164 163 162
161 160 159 158 157 156 155 154 153 152 151 150 149 148 147 146 145 144 143 142
141 140 139 138 137 136 135 134 133 132 131 130 129 128 127 126 125 124 123 122
121 120 119 118 117 116 115 114 113 112 111 110 109 108 107 106 105 104 103 102
101 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75
74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47
46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19
18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

attributes:

4 11 13 15 17 18 20 24 26 32 46 49 51 54 55 56

objects:

4026 4025 4024 4023 4022 4021 4020 4019 4018 4017 4016 4015 4014 4013 4012 4011
4010 4009 4008 4007 4006 4005 4004 4003 4002 4001 4000 3999 3998 3997 3996 3995
3994 3993 3992 3991 3990 3989 3988 3987 3986 3985 3984 3983 3982 3981 3980 3979
3978 3977 3976 3975 3974 3973 3972 3971 3970 3969 3968 3967 3966 3965 3964 3963
3962 3961 3960 3959 3958 3957 3956 3955 3954 3953 3952 3951 3950 3949 3948 3947

3946 3945 3944 3943 3942 3941 3940 3939 3938 3937 3936 3935 3934 3933 3932 3931
3930 3929 3928 3927 3926 3925 3924 3923 3922 3921 3920 3919 3918 3917 3916 3915
3914 3913 3912 3911 3910 3909 3908 3907 3906 3905 3904 3903 3902 3901 3900 3899
3898 3897 3896 3895 3894 3893 3892 3891 3890 3889 3888 3887 3886 3885 3884 3883
3882 3881 3880 3879 3878 3877 3876 3875 3874 3873 3872 3871 3870 3869 3868 3867
3866 3865 3864 3863 3862 3861 3860 3859 3858 3857 3856 3855 3854 3853 3852 3851
3850 3849 3848 3847 3846 3845 3844 3843 3842 3841 3840 3839 3838 3837 3836 3835
3834 3833 3832 3831 3830 3829 3828 3827 3826 3825 3824 3823 3822 3821 3820 3819
3818 3817 3816 3815 3814 3813 3812 3811 3810 3809 3808 3807 3806 3805 3804 3803
3802 3801 3800 3799 3798 3797 3796 3795 3794 3793 3792 3791 3790 3789 3788 3787
3786 3785 3784 3783 3782 3781 3780 3779 3778 3777 3776 3775 3774 3773 3772 3771
3770 3769 3768 3767 3766 3765 3764 3763 3762 3761 3760 3759 3758 3757 3756 3755
3754 3753 3752 3751 3750 3749 3748 3747 3746 3745 3744 3743 3742 3741 3740 3739
3738 3737 3736 3735 3734 3733 3732 3731 3730 3729 3728 3727 3726 3725 3724 3723
3722 3721 3720 3719 3718 3717 3716 3715 3714 3713 3712 3711 3710 3709 3708 3707
3706 3705 3704 3703 3702 3701 3700 3699 3698 3697 3696 3695 3694 3693 3692 3691
3690 3689 3688 3687 3686 3685 3684 3683 3682 3681 3680 3679 3678 3677 3676 3675
3674 3673 3672 3671 3669 3668 3667 3666 3665 3664 3663 3662 3661 3660 3659 3658
3657 3656 3655 3654 3653 3652 3651 3650 3649 3648 3647 3646 3645 3644 3643 3642
3641 3640 3639 3638 3637 3636 3635 3634 3633 3632 3631 3630 3629 3628 3627 3626
3625 3624 3623 3622 3621 3620 3619 3618 3617 3616 3615 3614 3613 3612 3611 3610
3609 3608 3607 3606 3605 3604 3603 3602 3601 3600 3599 3598 3597 3596 3595 3594
3593 3592 3591 3590 3589 3588 3587 3586 3585 3584 3583 3582 3581 3580 3579 3578
3577 3576 3575 3574 3573 3572 3571 3570 3569 3568 3567 3566 3565 3564 3563 3562
3561 3560 3559 3558 3557 3556 3555 3554 3553 3552 3551 3550 3549 3548 3547 3546
3545 3544 3543 3542 3541 3540 3539 3538 3537 3536 3535 3534 3533 3532 3531 3530
3529 3528 3527 3526 3525 3524 3523 3522 3521 3520 3519 3518 3517 3516 3515 3514
3513 3512 3511 3510 3509 3508 3507 3506 3505 3504 3503 3502 3501 3500 3499 3498
3497 3496 3495 3494 3493 3492 3491 3490 3489 3488 3487 3486 3485 3484 3483 3482
3481 3480 3479 3478 3477 3476 3475 3474 3473 3472 3471 3470 3469 3468 3467 3466
3465 3464 3463 3462 3461 3460 3459 3458 3457 3456 3455 3454 3453 3452 3451 3450
3449 3448 3447 3446 3445 3444 3443 3442 3441 3440 3439 3438 3437 3436 3435 3434
3433 3432 3431 3430 3429 3428 3427 3426 3425 3424 3423 3422 3421 3420 3419 3418
3417 3416 3415 3414 3413 3412 3411 3410 3409 3408 3407 3406 3405 3404 3403 3402
3401 3400 3399 3398 3397 3396 3395 3394 3393 3392 3391 3390 3389 3388 3387 3386
3385 3384 3383 3382 3381 3380 3379 3378 3377 3376 3375 3374 3373 3372 3371 3370

3369 3368 3367 3366 3365 3364 3363 3362 3361 3360 3359 3358 3357 3356 3355 3354
3353 3352 3351 3350 3349 3348 3347 3346 3345 3344 3343 3342 3341 3340 3339 3338
3337 3336 3335 3334 3333 3332 3331 3330 3329 3328 3327 3326 3325 3324 3323 3322
3321 3320 3319 3318 3317 3316 3315 3314 3313 3312 3311 3310 3309 3308 3307 3306
3305 3304 3303 3302 3301 3300 3299 3298 3297 3296 3295 3294 3293 3292 3291 3290
3289 3288 3287 3286 3285 3284 3283 3282 3281 3280 3279 3278 3277 3276 3275 3274
3273 3272 3271 3270 3269 3268 3267 3266 3265 3264 3263 3262 3261 3260 3259 3258
3257 3256 3255 3254 3253 3252 3251 3250 3249 3248 3247 3246 3245 3244 3243 3242
3241 3240 3239 3238 3237 3236 3235 3234 3233 3232 3231 3230 3229 3228 3227 3226
3225 3224 3223 3222 3221 3220 3219 3218 3217 3216 3215 3214 3213 3212 3211 3210
3209 3208 3207 3206 3205 3204 3203 3202 3201 3200 3199 3198 3197 3196 3195 3194
3193 3192 3191 3190 3189 3188 3187 3186 3185 3184 3183 3182 3181 3180 3179 3178
3177 3176 3175 3174 3173 3172 3171 3170 3169 3168 3167 3166 3165 3164 3163 3162
3161 3160 3159 3158 3157 3156 3155 3154 3153 3152 3151 3150 3149 3148 3147 3146
3145 3144 3143 3142 3141 3140 3139 3138 3137 3136 3135 3134 3133 3132 3131 3130
3129 3128 3127 3126 3125 3124 3123 3122 3121 3120 3119 3118 3117 3116 3115 3114
3113 3112 3111 3110 3109 3108 3107 3106 3105 3104 3103 3102 3101 3100 3099 3098
3097 3096 3095 3094 3093 3092 3091 3090 3089 3088 3087 3086 3085 3084 3083 3082
3081 3080 3079 3078 3077 3076 3075 3074 3073 3072 3071 3070 3069 3068 3067 3066
3065 3064 3063 3062 3061 3060 3059 3058 3057 3056 3055 3054 3053 3052 3051 3050
3049 3048 3047 3046 3045 3044 3043 3042 3041 3040 3039 3038 3037 3036 3035 3034
3033 3032 3031 3030 3029 3028 3027 3026 3025 3024 3023 3022 3021 3020 3019 3018
3017 3016 3015 3014 3013 3012 3011 3010 3009 3008 3007 3006 3005 3004 3003 3002
3001 3000 2999 2998 2997 2996 2995 2994 2993 2992 2991 2990 2989 2988 2987 2986
2985 2984 2983 2982 2981 2980 2979 2978 2977 2976 2975 2974 2973 2972 2971 2970
2969 2968 2967 2966 2965 2964 2963 2962 2961 2960 2959 2958 2957 2956 2955 2954
2953 2952 2951 2950 2949 2948 2947 2946 2945 2944 2943 2942 2941 2940 2939 2938
2937 2936 2935 2934 2933 2932 2931 2930 2929 2928 2927 2926 2925 2924 2923 2922
2921 2920 2919 2918 2917 2916 2915 2914 2913 2912 2911 2910 2909 2908 2907 2906
2905 2904 2903 2902 2901 2900 2899 2898 2897 2896 2895 2894 2893 2892 2891 2890
2889 2888 2887 2886 2885 2884 2883 2882 2881 2880 2879 2878 2877 2876 2875 2874
2873 2872 2871 2870 2869 2868 2867 2866 2865 2864 2863 2862 2861 2860 2859 2858
2857 2856 2855 2854 2853 2852 2851 2850 2849 2848 2847 2846 2845 2844 2843 2842
2841 2840 2839 2838 2837 2836 2835 2834 2833 2832 2831 2830 2829 2828 2827 2826
2825 2824 2823 2822 2821 2820 2819 2818 2817 2816 2815 2814 2813 2812 2811 2810
2809 2808 2807 2806 2805 2804 2803 2802 2801 2800 2799 2798 2797 2796 2795 2794

2793 2792 2791 2790 2789 2788 2787 2786 2785 2784 2783 2782 2781 2780 2779 2778
2777 2776 2775 2774 2773 2772 2771 2770 2769 2768 2767 2766 2765 2764 2763 2762
2761 2760 2759 2758 2757 2756 2755 2754 2753 2752 2751 2750 2749 2748 2747 2746
2745 2744 2743 2742 2741 2740 2739 2738 2737 2736 2735 2734 2733 2732 2731 2730
2729 2728 2727 2726 2725 2724 2723 2722 2721 2720 2719 2718 2717 2716 2715 2714
2713 2712 2711 2710 2709 2708 2707 2706 2705 2704 2703 2702 2701 2700 2699 2698
2697 2696 2695 2694 2693 2692 2691 2690 2689 2688 2687 2686 2685 2684 2683 2682
2681 2680 2679 2678 2677 2676 2675 2674 2673 2672 2671 2670 2669 2668 2667 2666
2665 2664 2663 2662 2661 2660 2659 2658 2657 2656 2655 2654 2653 2652 2651 2650
2649 2648 2647 2646 2645 2644 2643 2642 2641 2640 2639 2638 2637 2636 2635 2634
2633 2632 2631 2630 2629 2628 2627 2626 2625 2624 2623 2622 2621 2620 2619 2618
2617 2616 2615 2614 2613 2612 2611 2610 2609 2608 2607 2606 2605 2604 2603 2602
2601 2600 2599 2598 2597 2596 2595 2594 2593 2592 2591 2590 2589 2588 2587 2586
2585 2584 2583 2582 2581 2580 2579 2578 2577 2576 2575 2574 2573 2572 2571 2570
2569 2568 2567 2566 2565 2564 2563 2562 2561 2560 2559 2558 2557 2556 2555 2554
2553 2552 2551 2550 2549 2548 2547 2546 2545 2544 2543 2542 2541 2540 2539 2538
2537 2536 2535 2534 2533 2532 2531 2530 2529 2528 2527 2526 2525 2524 2523 2522
2521 2520 2519 2518 2517 2516 2515 2514 2513 2512 2511 2510 2509 2508 2507 2506
2505 2504 2503 2502 2501 2500 2499 2498 2497 2496 2495 2494 2493 2492 2491 2490
2489 2488 2487 2486 2485 2484 2483 2482 2481 2480 2479 2478 2477 2476 2475 2474
2473 2472 2471 2470 2469 2468 2467 2466 2465 2464 2463 2462 2461 2460 2459 2458
2457 2456 2455 2454 2453 2452 2451 2450 2449 2448 2447 2446 2445 2444 2443 2442
2441 2440 2439 2438 2437 2436 2435 2434 2433 2432 2431 2430 2429 2428 2427 2426
2425 2424 2423 2422 2421 2420 2419 2418 2417 2416 2415 2414 2413 2412 2411 2410
2409 2408 2407 2406 2405 2404 2403 2402 2401 2400 2399 2398 2397 2396 2395 2394
2393 2392 2391 2390 2389 2388 2387 2386 2385 2384 2383 2382 2381 2380 2379 2378
2377 2376 2375 2374 2373 2372 2371 2370 2369 2368 2367 2366 2365 2364 2363 2362
2361 2360 2359 2358 2357 2356 2355 2354 2353 2352 2351 2350 2349 2348 2347 2346
2345 2344 2343 2342 2341 2340 2339 2338 2337 2336 2335 2334 2333 2332 2331 2330
2329 2328 2327 2326 2325 2324 2323 2322 2321 2320 2319 2318 2317 2316 2315 2314
2313 2312 2311 2310 2309 2308 2307 2306 2305 2304 2303 2302 2301 2300 2299 2298
2297 2296 2295 2294 2293 2292 2291 2290 2289 2288 2287 2286 2285 2284 2283 2282
2281 2280 2279 2278 2277 2276 2275 2274 2273 2272 2271 2270 2269 2268 2267 2266
2265 2264 2263 2262 2261 2260 2259 2258 2257 2256 2255 2254 2253 2252 2251 2250
2249 2248 2247 2246 2245 2244 2243 2242 2241 2240 2239 2238 2237 2236 2235 2234
2233 2232 2231 2230 2229 2228 2227 2226 2225 2224 2223 2222 2221 2220 2219 2218

2217 2216 2215 2214 2213 2212 2211 2210 2209 2208 2207 2206 2205 2204 2203 2202
2201 2200 2199 2198 2197 2196 2195 2194 2193 2192 2191 2190 2189 2188 2187 2186
2185 2184 2183 2182 2181 2180 2179 2178 2177 2176 2175 2174 2173 2172 2171 2170
2169 2168 2167 2166 2165 2164 2163 2162 2161 2160 2159 2158 2157 2156 2155 2154
2153 2152 2151 2150 2149 2148 2147 2146 2145 2144 2143 2142 2141 2140 2139 2138
2137 2136 2135 2134 2133 2132 2131 2130 2129 2128 2127 2126 2125 2124 2123 2122
2121 2120 2119 2118 2117 2116 2115 2114 2113 2112 2111 2110 2109 2108 2107 2106
2105 2104 2103 2102 2101 2100 2099 2098 2097 2096 2095 2094 2093 2092 2091 2090
2089 2088 2087 2086 2085 2084 2083 2082 2081 2080 2079 2078 2077 2076 2075 2074
2073 2072 2071 2070 2069 2068 2067 2066 2065 2064 2063 2062 2061 2060 2059 2058
2057 2056 2055 2054 2053 2052 2051 2050 2049 2048 2047 2046 2045 2044 2043 2042
2041 2040 2039 2038 2037 2036 2035 2034 2033 2032 2031 2030 2029 2028 2027 2026
2025 2024 2023 2022 2021 2020 2019 2018 2017 2016 2015 2014 2013 2012 2011 2010
2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1998 1997 1996 1995 1994
1993 1992 1991 1990 1989 1988 1987 1986 1985 1984 1983 1982 1981 1980 1979 1978
1977 1976 1975 1974 1973 1972 1971 1970 1969 1968 1967 1966 1965 1964 1963 1962
1961 1960 1959 1958 1957 1956 1955 1954 1953 1952 1951 1950 1949 1948 1947 1946
1945 1944 1943 1942 1941 1940 1939 1938 1937 1936 1935 1934 1933 1932 1931 1930
1929 1928 1927 1926 1925 1924 1923 1922 1921 1920 1919 1918 1917 1916 1915 1914
1913 1912 1911 1910 1909 1908 1907 1906 1905 1904 1903 1902 1901 1900 1899 1898
1897 1896 1895 1894 1893 1892 1891 1890 1889 1888 1887 1886 1885 1884 1883 1882
1881 1880 1879 1878 1877 1876 1875 1874 1873 1872 1871 1870 1869 1868 1867 1866
1865 1864 1863 1862 1861 1860 1859 1858 1857 1856 1855 1854 1853 1852 1851 1850
1849 1848 1847 1846 1845 1844 1843 1842 1841 1840 1839 1838 1837 1836 1835 1834
1833 1832 1831 1830 1829 1828 1827 1826 1825 1824 1823 1822 1821 1820 1819 1818
1817 1816 1815 1814 1813 1812 1811 1810 1809 1808 1807 1806 1805 1804 1803 1802
1801 1800 1799 1798 1797 1796 1795 1794 1793 1792 1791 1790 1789 1788 1787 1786
1785 1784 1783 1782 1781 1780 1779 1778 1777 1776 1775 1774 1773 1772 1771 1770
1769 1768 1767 1766 1765 1764 1763 1762 1761 1760 1759 1758 1757 1756 1755 1754
1753 1752 1751 1750 1749 1748 1747 1746 1745 1744 1743 1742 1741 1740 1739 1738
1737 1736 1735 1734 1733 1732 1731 1730 1729 1728 1727 1726 1725 1724 1723 1722
1721 1720 1719 1718 1717 1716 1715 1714 1713 1712 1711 1710 1709 1708 1707 1706
1705 1704 1703 1702 1701 1700 1699 1698 1697 1696 1695 1694 1693 1692 1691 1690
1689 1688 1687 1686 1685 1684 1683 1682 1681 1680 1679 1678 1677 1676 1675 1674
1673 1672 1671 1670 1669 1668 1667 1666 1665 1664 1663 1662 1661 1660 1659 1658
1657 1656 1655 1654 1653 1652 1651 1650 1649 1648 1647 1646 1645 1644 1643 1642

1641 1640 1639 1638 1637 1636 1635 1634 1633 1632 1631 1630 1629 1628 1627 1626
1625 1624 1623 1622 1621 1620 1619 1618 1617 1616 1615 1614 1613 1612 1611 1610
1609 1608 1607 1606 1605 1604 1603 1602 1601 1600 1599 1598 1597 1596 1595 1594
1593 1592 1591 1590 1589 1588 1587 1586 1585 1584 1583 1582 1581 1580 1579 1578
1577 1576 1575 1574 1573 1572 1571 1570 1569 1568 1567 1566 1565 1564 1563 1562
1561 1560 1559 1558 1557 1556 1555 1554 1553 1552 1551 1550 1549 1548 1547 1546
1545 1544 1543 1542 1541 1540 1539 1538 1537 1536 1535 1534 1533 1532 1531 1530
1529 1528 1527 1526 1525 1524 1523 1522 1521 1520 1519 1518 1517 1516 1515 1514
1513 1512 1511 1510 1509 1508 1507 1506 1505 1504 1503 1502 1501 1500 1499 1498
1497 1496 1495 1494 1493 1492 1491 1490 1489 1488 1487 1486 1485 1484 1483 1482
1481 1480 1479 1478 1477 1476 1475 1474 1473 1472 1471 1470 1469 1468 1467 1466
1465 1464 1463 1462 1461 1460 1459 1458 1457 1456 1455 1454 1453 1452 1451 1450
1449 1448 1447 1446 1445 1444 1443 1442 1441 1440 1439 1438 1437 1436 1435 1434
1433 1432 1431 1430 1429 1428 1427 1426 1425 1424 1423 1422 1421 1420 1419 1418
1417 1416 1415 1414 1413 1412 1411 1410 1409 1408 1407 1406 1405 1404 1403 1402
1401 1400 1399 1398 1397 1396 1395 1394 1393 1392 1391 1390 1389 1388 1387 1386
1385 1384 1383 1382 1381 1380 1379 1378 1377 1376 1375 1374 1373 1372 1371 1370
1369 1368 1367 1366 1365 1364 1363 1362 1361 1360 1359 1358 1357 1356 1355 1354
1353 1352 1351 1350 1349 1348 1347 1346 1345 1344 1343 1342 1341 1340 1339 1338
1337 1336 1335 1334 1333 1332 1331 1330 1329 1328 1327 1326 1325 1324 1323 1322
1321 1320 1319 1318 1317 1316 1315 1314 1313 1312 1311 1310 1309 1308 1307 1306
1305 1304 1303 1302 1301 1300 1299 1298 1297 1296 1295 1294 1293 1292 1291 1290
1289 1288 1287 1286 1285 1284 1283 1282 1281 1280 1279 1278 1277 1276 1275 1274
1273 1272 1271 1270 1269 1268 1267 1266 1265 1264 1263 1262 1261 1260 1259 1258
1257 1256 1255 1254 1253 1252 1251 1250 1249 1248 1247 1246 1245 1244 1243 1242
1241 1240 1239 1238 1237 1236 1235 1234 1233 1232 1231 1230 1229 1228 1227 1226
1225 1224 1223 1222 1221 1220 1219 1218 1217 1216 1215 1214 1213 1212 1211 1210
1209 1208 1207 1206 1205 1204 1203 1202 1201 1200 1199 1198 1197 1196 1195 1194
1193 1192 1191 1190 1189 1188 1187 1186 1185 1184 1183 1182 1181 1180 1179 1178
1177 1176 1175 1174 1173 1172 1171 1170 1169 1168 1167 1166 1165 1164 1163 1162
1161 1160 1159 1158 1157 1156 1155 1154 1153 1152 1151 1150 1149 1148 1147 1146
1145 1144 1143 1142 1141 1140 1139 1138 1137 1136 1135 1134 1133 1132 1131 1130
1129 1128 1127 1126 1125 1124 1123 1122 1121 1120 1119 1118 1117 1116 1115 1114
1113 1112 1111 1110 1109 1108 1107 1106 1105 1104 1103 1102 1101 1100 1099 1098
1097 1096 1095 1094 1093 1092 1091 1090 1089 1088 1087 1086 1085 1084 1083 1082
1081 1080 1079 1078 1077 1076 1075 1074 1073 1072 1071 1070 1069 1068 1067 1066

1065 1064 1063 1062 1061 1060 1059 1058 1057 1056 1055 1054 1053 1052 1051 1050
1049 1048 1047 1046 1045 1044 1043 1042 1041 1040 1039 1038 1037 1036 1035 1034
1033 1032 1030 1029 1028 1027 1026 1025 1024 1023 1022 1021 1020 1019 1018 1017
1016 1015 1014 1013 1012 1011 1010 1009 1008 1007 1006 1005 1004 1003 1002 1001
1000 999 998 997 996 995 994 993 992 991 990 989 988 987 986 985 984 983 982 981
980 979 978 977 976 975 974 973 972 971 970 969 968 967 966 965 964 963 962 961
960 959 958 957 956 955 954 953 952 951 950 949 948 947 946 945 944 943 942 941
940 939 938 937 936 935 934 933 932 931 930 929 928 927 926 925 924 923 922 921
920 919 918 917 916 915 914 913 912 911 910 909 908 907 906 905 904 903 902 901
900 899 898 897 896 895 894 893 892 891 890 889 888 887 886 885 884 883 882 881
880 879 878 877 876 875 874 873 872 871 870 869 868 867 866 865 864 863 862 861
860 859 858 857 856 855 854 853 852 851 850 849 848 847 846 845 844 843 842 841
840 839 838 837 836 835 834 833 832 831 830 829 828 827 826 825 824 823 822 821
820 819 818 817 816 815 814 813 812 811 810 809 808 807 806 805 804 803 802 801
800 799 798 797 796 795 794 793 792 791 790 789 788 787 786 785 784 783 782 781
780 779 778 777 776 775 774 773 772 771 770 769 768 767 766 765 764 763 762 761
760 759 758 757 756 755 754 753 752 751 750 749 748 747 746 745 744 743 742 741
740 739 738 737 736 735 734 733 732 731 730 729 728 727 726 725 724 723 722 721
720 719 718 717 716 715 714 713 712 711 710 709 708 707 706 705 704 703 702 701
700 699 698 697 696 695 694 693 692 691 690 689 688 687 686 685 684 683 682 681
680 679 678 677 676 675 674 673 672 671 670 669 668 667 666 665 664 663 662 661
660 659 658 657 656 655 654 653 652 651 650 649 648 647 646 645 644 643 642 641
640 639 638 637 636 635 634 633 632 631 630 629 628 627 626 625 624 623 622 621
620 619 618 617 616 615 614 613 612 611 610 609 608 607 606 605 604 603 602 601
600 599 598 597 596 595 594 593 592 591 590 589 588 587 586 585 584 583 582 581
580 579 578 577 576 575 574 573 572 571 570 569 568 567 566 565 564 563 562 561
560 559 558 557 556 555 554 553 552 551 550 549 548 547 546 545 544 543 542 541
540 539 538 537 536 535 534 533 532 531 530 529 528 527 526 525 524 523 522 521
520 519 518 517 516 515 514 513 512 511 510 509 508 507 506 505 504 503 502 501
500 499 498 497 496 495 494 493 492 491 490 489 488 487 486 485 484 483 482 481
480 479 478 477 476 475 474 473 472 471 470 469 468 467 466 465 464 463 462 461
460 459 458 457 456 455 454 453 452 451 450 449 448 447 446 445 444 443 442 441
440 439 438 437 436 435 434 433 432 431 430 429 428 427 426 425 424 423 422 421
420 419 418 417 416 415 414 413 412 411 410 409 408 407 406 405 404 403 402 401
400 399 398 397 396 395 394 393 392 391 390 389 388 387 386 385 384 383 382 381
380 379 378 377 376 375 374 373 372 371 370 369 368 367 366 365 364 363 362 361

360 359 358 357 356 355 354 353 352 351 350 349 348 347 346 345 344 343 342 341
340 339 338 337 336 335 334 333 332 331 330 329 328 327 326 325 324 323 322 321
320 319 318 317 316 315 314 313 312 311 310 309 308 307 306 305 304 303 302 301
300 299 298 297 296 295 294 293 292 291 290 289 288 287 286 285 284 283 282 281
280 279 278 277 276 275 274 273 272 271 270 269 268 267 266 265 264 263 262 261
260 259 258 257 256 255 254 253 252 251 250 249 248 247 246 245 244 243 242 241
240 239 238 237 236 235 234 233 232 231 230 229 228 227 226 225 224 223 222 221
220 219 218 217 216 215 214 213 212 211 210 209 208 207 206 205 204 203 202 201
200 199 198 197 196 195 194 193 192 191 190 189 188 187 186 185 184 183 182 181
180 179 178 177 176 175 174 173 172 171 170 169 168 167 166 165 164 163 162 161
160 159 158 157 156 155 154 153 152 151 150 149 148 147 146 145 144 143 142 141
140 139 138 137 136 135 134 133 132 131 130 129 128 127 126 125 124 123 122 121
120 119 118 117 116 115 114 113 112 111 110 109 108 107 106 105 104 103 102 101
100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74
73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46
45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18
17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Breast Cancer Wisconsin (Diagnostic) dataset:

attributes:

7 16 19 22 23 26 28 30 31

objects:

475 474 473 472 471 470 469 468 467 466 465 464 463 462 460 459 458 457 456 455
454 453 452 451 450 449 448 447 446 445 444 443 442 441 440 439 438 437 436 435
434 433 432 431 430 429 428 427 426 425 424 423 422 421 420 419 418 417 416 415
414 413 412 411 410 409 408 407 406 405 404 403 402 401 400 399 398 397 396 395
394 393 392 391 390 389 388 387 386 385 384 383 382 381 380 379 378 377 376 375
374 373 372 371 370 369 368 367 366 365 364 363 362 361 360 359 358 357 356 355
354 353 352 351 350 349 348 347 346 345 344 343 342 341 340 339 338 337 336 335
334 333 332 331 330 329 328 327 326 325 324 323 322 321 320 319 318 317 316 315
314 313 312 311 310 309 308 307 306 305 304 303 302 301 300 299 298 297 296 295
294 293 292 291 290 289 288 287 286 285 284 283 282 281 280 279 278 277 276 275
274 273 272 271 270 269 268 267 266 265 264 263 262 261 260 259 258 257 256 255
254 253 252 251 250 249 248 247 246 245 244 243 242 241 240 239 238 237 236 235
234 233 232 231 230 229 228 227 226 225 224 223 222 221 220 219 218 217 216 215
214 213 212 211 210 209 208 207 206 205 204 203 202 201 200 199 198 197 196 195
194 193 192 191 190 189 188 187 186 185 184 183 182 181 180 179 178 177 176 175

174 173 172 171 170 169 168 167 166 165 164 163 162 161 160 159 158 157 156 155 154 153 152 151 150 149 148 147 146 145 144 143 142 141 140 139 138 137 136 135 134 133 132 131 130 129 128 127 126 125 124 123 122 121 120 119 118 117 116 115 114 113 112 111 110 109 108 107 106 105 104 103 102 101 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

attributes:

2 3 9 19 24 25 26 27 29 30

objects:

475 474 473 472 471 470 469 468 467 466 465 464 463 462 461 460 459 458 457 456 455 454 453 452 451 450 449 448 447 446 445 444 443 442 441 440 439 438 437 436 435 434 433 432 431 430 429 428 427 426 425 424 423 422 421 420 419 418 417 416 415 414 413 412 411 410 409 408 407 406 405 404 403 402 401 400 399 398 397 396 395 394 393 392 391 390 389 388 387 386 385 384 383 382 381 380 379 378 377 376 375 374 373 372 371 370 369 368 367 366 365 364 363 362 361 360 359 358 357 356 355 354 353 352 351 350 349 348 347 346 345 344 343 342 341 340 339 338 337 336 335 334 333 332 331 330 329 328 327 326 325 324 323 322 321 320 319 318 317 316 315 314 313 312 311 310 309 308 307 306 305 304 303 302 301 300 299 298 297 296 295 294 293 292 291 290 289 288 287 286 285 284 283 282 281 280 279 278 277 276 275 274 273 272 271 270 269 268 267 266 265 264 263 262 261 260 259 258 257 256 255 254 253 252 251 250 249 248 247 246 245 244 243 242 241 240 239 238 237 236 235 234 233 232 231 230 229 228 227 226 225 224 223 222 221 220 219 218 217 216 215 214 213 212 211 210 209 208 207 206 205 204 203 202 201 200 199 198 197 196 195 194 193 192 191 190 189 188 187 186 185 184 183 182 181 180 179 178 177 176 175 174 173 172 171 170 169 168 167 166 165 164 163 162 161 160 159 158 157 156 155 154 153 152 151 150 149 148 147 146 145 144 143 142 141 140 139 138 137 136 135 134 133 132 131 130 129 128 127 126 125 124 123 122 121 120 119 118 117 116 115 114 113 112 111 110 109 108 107 106 105 104 103 102 101 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

attributes:

3 5 8 13 14 18 21 24 26 28

objects:

475 474 473 472 471 470 469 468 467 466 465 464 463 462 461 460 459 458 457 456
455 454 453 452 451 450 449 448 447 446 445 444 443 442 441 440 439 438 437 436
435 434 433 432 431 430 429 428 427 426 425 424 423 422 421 420 419 418 417 416
415 414 413 412 411 410 409 408 407 406 405 404 403 402 401 400 399 398 397 396
395 394 393 392 391 390 389 388 387 386 385 384 383 382 381 380 379 378 377 376
375 374 373 372 371 370 369 368 367 366 365 364 363 362 361 360 359 358 357 356
355 354 353 352 351 350 349 348 347 346 345 344 343 342 341 340 339 338 337 336
335 334 333 332 331 330 329 328 327 326 325 324 323 322 321 320 319 318 317 316
315 314 313 312 311 310 309 308 307 306 305 304 303 302 301 300 299 298 297 296
295 294 293 292 291 290 289 288 287 286 285 284 283 282 281 280 279 278 277 276
275 274 273 272 271 270 269 268 267 266 265 264 263 262 261 260 259 258 257 256
255 254 253 252 251 250 249 248 247 246 245 244 243 242 241 240 239 238 237 236
235 234 233 232 231 230 229 228 227 226 225 224 223 222 221 220 219 218 217 216
215 214 213 212 211 210 209 208 207 206 205 204 203 202 201 200 199 198 197 196
195 194 193 192 191 190 189 188 187 186 185 184 183 182 181 180 179 178 177 176
175 174 173 172 171 170 169 168 167 166 165 164 163 162 161 160 159 158 157 156
155 154 153 152 151 150 149 148 147 146 145 144 143 142 141 140 139 138 137 136
135 134 133 132 131 130 129 128 127 126 125 124 123 122 121 120 119 118 117 116
115 114 113 112 111 110 109 108 107 106 105 104 103 102 101 100 99 98 97 96 95 94
93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66
65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38
37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10
9 8 7 6 5 4 3 2 1

SPECT dataset:

attributes:

2 3 4 8 10 11 13 16 19 20 22

objects:

234 232 231 230 229 228 227 225 224 223 222 221 220 219 218 217 216 215 214 213
212 211 210 209 208 207 206 205 204 203 202 201 200 199 198 197 196 195 194 193
192 191 190 188 187 186 185 184 183 182 181 180 179 178 177 176 175 174 173 172
171 169 168 167 166 165 164 163 162 161 160 159 158 157 156 155 154 152 151 150
149 148 147 146 145 144 143 142 141 140 139 138 137 136 135 134 133 132 131 130
129 128 126 125 124 123 122 121 120 119 118 117 116 115 114 113 112 111 109 108
107 105 104 103 102 101 100 99 98 97 96 95 94 93 92 91 89 88 87 86 85 84 83 82 81
80 79 78 77 76 75 74 72 71 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51

50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 31 29 28 27 26 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

attributes:

1 2 3 4 9 10 11 12 13 16 19 20 22

objects:

234 232 231 230 229 228 227 225 224 223 222 221 220 219 218 217 216 215 214 213 212 211 210 209 208 207 206 205 204 203 202 201 200 199 198 197 196 195 194 193 192 191 190 188 187 186 185 184 183 182 181 180 179 178 177 176 175 174 173 172 171 169 168 167 166 165 164 163 162 161 160 159 158 157 156 155 154 153 152 151 150 149 148 147 146 145 144 143 142 141 140 139 138 137 136 135 134 133 132 131 130 129 128 127 126 125 124 123 122 121 120 119 118 117 115 114 113 112 111 109 108 107 106 105 104 103 102 101 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 40 39 38 36 35 34 33 31 29 28 27 26 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

attributes:

1 4 8 13 16 17 19 20 21 22

objects:

234 232 231 230 229 228 227 225 224 223 222 221 220 219 218 217 216 215 214 213 212 211 210 209 208 207 206 205 204 203 202 201 200 198 197 196 195 194 193 192 191 190 188 187 186 185 184 183 182 181 180 179 178 177 176 175 174 173 172 171 169 168 167 166 165 164 163 162 161 160 159 158 157 155 154 152 151 150 149 148 146 145 144 143 142 141 140 139 138 137 136 135 134 133 132 131 130 129 128 126 125 124 123 122 121 120 119 118 117 116 115 114 113 112 111 109 108 107 105 104 103 102 101 100 99 98 97 96 95 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 72 71 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 31 29 28 27 26 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1